



Operating Systems and Languages Library

MS-DOS

User Guide



olivetti
PERSONAL
COMPUTER



olivetti



PREFACE

This manual is a user guide for the MS-DOS operating system. It describes Microsoft Version 3.20 of MS-DOS, and is for anyone who wishes to use this operating system on an Olivetti Personal Computer.

SUMMARY

The first chapter provides a general introduction to MS-DOS.

Chapters 2 and 3 describe in more detail the major functions and features of MS-DOS. These include: using control keys and function keys; files and the hierarchical directory structure; entering and using MS-DOS commands.

Chapter 5 is a full and detailed reference to all the MS-DOS commands. It includes a section on commands to use and those not to use when the computer is connected to a network.

Chapters 6 to 10 provide detailed reference information on the Video File Editor (EDIT), the Line Editor (EDLIN), the Linker (MS-LINK) and the Debugger (DEBUG).

RELATED PUBLICATIONS

Installation and Operations Guide for your Personal Computer
Getting Started With MS-DOS: Software Installation Guide
(Code 4040360 J)

MS GW-BASIC Interpreter under MS-DOS User Guide
(Code 4034490 C)

MS-DOS System Programmer Guide, Vol I (Code 4024270 M)
MS-DOS System Programmer Guide, Vol II (Code 4033300 G)
MS-DOS Quick Reference Guide (Code 4034470 S)

DISTRIBUTION: General (G)

FIRST EDITION: January 1987

*Copyright © Microsoft Corporation
1980 - 1986*

*Copyright © 1987 by Olivetti
All rights reserved*

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C., S.p.A.
Direzione Documentazione
77, Via Jervis - 10015 Ivrea — Italy

- OLIVETTI is a trademark of Ing. C. Olivetti & C., S.p.A.
- OLITALK is a trademark of Ing. C. Olivetti & C., S.p.A.
- OLITERM is a trademark of Ing. C. Olivetti & C., S.p.A.
- SORTP is a trademark of Ing. C. Olivetti & C., S.p.A.
- ASM-86 is a trademark of Digital Research
- CB-86 is a trademark of Digital Research
- CBASIC-86 is a trademark of Digital Research
- CLEO is a trademark of Phone 1 Inc.
- Concurrent CP/M-86 is a trademark of Digital Research
- Concurrent DOS is a trademark of Digital Research
- CP/M-86 is a trademark of Digital Research
- DDT-86 is a trademark of Digital Research
- Dr. Draw is a trademark of Digital Research
- Dr. Graph is a trademark of Digital Research
- Dr. Logo is a trademark of Digital Research
- ETHERNET is a trademark of Xerox Corp.
- GEM COLLECTION is a trademark of Digital Research
- GEM DESKTOP is a trademark of Digital Research
- GEM DRAW is a trademark of Digital Research
- GEM GRAPH is a trademark of Digital Research
- GEM PAINT is a trademark of Digital Research
- GEM WORDCHART is a trademark of Digital Research
- GEM WRITE is a trademark of Digital Research
- GSS is a registered trademark of Graphic Software Systems, Inc.
- GSS-CHART is a trademark of Graphic Software Systems, Inc.
- GSS-DRIVERS is a trademark of Graphic Software Systems, Inc.
- GSS-KERNEL SYSTEM is a trademark of Graphic Software Systems, Inc.
- GSS-TOOLKIT is a trademark of Graphic Software Systems, Inc.
- GSX-86 is a trademark of Digital Research
- GW is a trademark of Microsoft Corp.
- IBM is a registered trademark of International Business Machines Corp.
- MICROSOFT is a registered trademark of Microsoft Corp.
- MS is a trademark of Microsoft Corp.
- OMNINET is a trademark of Corvus Systems Inc.
- p-System is a trademark of Softech Microsystem, Inc
- PC-DOS is a trademark of International Business Machines Corp.
- PEACHPAK is a trademark of Peachtree Software International Ltd.
- Personal Basic is a trademark of Digital Research
- SID-86 is a trademark of Digital Research
- TerminAll is a trademark of TOPIC System, Inc.
- TOPIC is a trademark of TOPIC System, Inc.
- UCSD and UCSD Pascal are registered trademarks of the Regent of the University of California
- UNIX is a trademark of Bell Laboratories
- Z80 is a registered trademark of Zilog Inc.
- Z8000 is a registered trademark of Zilog Inc.
- XENIX is a registered trademark of Microsoft Corp.

CONTENTS

1. INTRODUCTION

WHAT IS MS-DOS?	1-1
COMMAND LIBRARY	1-1
FILE AND DIRECTORY HANDLING	1-1
PROGRAMMING TOOLS	1-1
INTERNAL AND EXTERNAL COMMANDS	1-2
BATCH PROCESSING	1-2
THE AUTOEXEC.BAT FILE	1-2
REAL-TIME CLOCK FUNCTIONS	1-2
INTERFACE HANDLING	1-3
DISKS	1-3
DISKETTE HANDLING	1-5
PURCHASING DISKETTES	1-6
LABELING DISKETTES	1-6
WRITE-PROTECTION	1-6
VIRTUAL DISK	1-8
NOTATION CONVENTIONS	1-9

2. MS-DOS CONTROL KEYS AND EDITING FUNCTION KEYS

INTRODUCTION	2-1
MS-DOS CONTROL KEYS	2-1
EDITING FUNCTION KEYS	2-4

3. FILES AND DIRECTORIES

HOW MS-DOS KEEPS TRACK OF YOUR FILES	3-1
HOW TO NAME YOUR FILES	3-1
WILD CARDS	3-2
THE ? WILD CARD	3-2
THE * WILD CARD	3-3
RESERVED DEVICE NAMES	3-4
HOW TO COPY YOUR FILES	3-4
HOW TO PROTECT YOUR FILES	3-5
HOW TO EXAMINE THE CONTENTS OF A FILE	3-6
DIRECTORIES	3-6
FILE NAMES AND PATHS	3-8
PATHNAMES	3-8
PATHS AND EXTERNAL COMMANDS	3-10
HOW TO DISPLAY YOUR CURRENT DIRECTORY	3-11
HOW TO CREATE A DIRECTORY	3-13
HOW TO CHANGE YOUR CURRENT DIRECTORY	3-13
HOW TO REMOVE A DIRECTORY	3-14
HOW TO CHECK THE VALIDITY OF YOUR FILES	3-14

4. ENTERING AND USING MS-DOS COMMANDS

COMMAND SYNTAX	4-1
PARAMETERS	4-1

INTERNAL AND EXTERNAL COMMANDS	4-6
EXTERNAL COMMAND SYNTAX	4-8
INFORMATION COMMON TO ALL MS-DOS COMMANDS	4-8
BATCH PROCESSING	4-10
HOW TO CREATE AND EXECUTE A BATCH FILE	4-10
THE AUTOEXEC.BAT FILE	4-13
BATCH FILES WITH REPLACEABLE PARAMETERS	4-14
INPUT AND OUTPUT	4-15
REDIRECTING YOUR OUTPUT	4-15
REDIRECTING YOUR INPUT	4-16
FILTERS	4-16
COMMAND PIPING	4-17

5. COMMANDS

INTRODUCTION	5-1
MS-DOS 3.20 AND NETWORKING	5-5
COMMANDS	5-8
ASSIGN	5-8
ATTRIB	5-10
BACKUP	5-12
BREAK	5-16
CHDIR	5-17
CHKDSK	5-19

CLS	5-23
COMMAND	5-23
COMP	5-27
COPY	5-30
CTTY	5-36
DATE	5-38
DEL	5-41
DIR	5-43
DISKCOMP	5-45
DISKCOPY	5-48
ECHO	5-51
ERASE	5-52
EXE2BIN	5-52
EXIT	5-55
FC	5-56
FDISK	5-66
FIND	5-66
FOR	5-70
FORMAT	5-72
GOTO	5-76
GRAFTABL	5-78
GRAPHICS	5-79

GWBasic	5-82
HEXDUMP	5-83
IF	5-85
JOIN	5-87
LABEL	5-92
MKDIR	5-94
MODE	5-95
MODE COM:	5-96
MODE n	5-98
MODE LPT	5-100
MORE	5-103
PATH	5-104
PAUSE	5-105
PRINT	5-107
PROMPT	5-111
RECOVER	5-114
REM	5-115
REN	5-116
REPLACE	5-118
RESTORE	5-122
RMDIR	5-126
SELECT	5-127

SET	5-129
SHARE	5-131
SHIFT	5-133
SORT	5-135
SUBST	5-139
SYS	5-143
TIME	5-144
TREE	5-146
TYPE	5-147
VER	5-149
VERIFY	5-149
VOL	5-150
XCOPY	5-151

6. VIDEO FILE EDITOR (EDIT)

INTRODUCTION	6-1
HOW TO INVOKE THE VIDEO FILE EDITOR	6-1
THE DISPLAY	6-3
THE KEYBOARD	6-4
EDIT	6-7
GENERAL EDITING FUNCTION KEYS	6-8
WINDOW MOVING FUNCTION KEYS	6-16
EXITING AND SAVING FUNCTION KEYS	6-18

COMMANDS AND SEARCHING	6-19
STRING SEARCHES	6-19
COMMANDS	6-21
GOTO	6-21
DELETE	6-22
FILE	6-22
 7. LINE EDITOR (EDLIN)	
INTRODUCTION	7-1
HOW TO INVOKE THE EDLIN PROGRAM	7-2
EDLIN	7-2
INTER-LINE COMMANDS	7-5
line (EDIT LINE)	7-6
A (APPEND LINES)	7-8
C (COPY LINES)	7-9
D (DELETE LINES)	7-12
E (END EDITING)	7-15
I (INSERT LINES)	7-16
L (LIST TEXT)	7-20
M (MOVE LINES)	7-23
P (PAGE)	7-25
Q (QUIT EDITING)	7-26
R (REPLACE TEXT)	7-27

S (SEARCH TEXT)	7-30
T (TRANSFER LINES)	7-34
W (WRITE LINES)	7-35
INTRA-LINE COMMANDS	7-36
COPY1	7-37
COPYTO	7-38
COPYLINE	7-40
SKIP1	7-41
SKIPTO	7-42
KILL	7-44
INS	7-46
NEWTEMP	7-47

8. THE LINKER

INTRODUCTION	8-1
TEMPORARY FILES	8-1
CHANGING DISKETTES	8-2
SEGMENTS, GROUPS AND CLASSES	8-2
INTERACTIVE ENTRY	8-3
COMMAND LINE ENTRY	8-10
AUTOMATIC RESPONSE FILE ENTRY	8-13

9. THE DEBUGGER

INTRODUCTION	9-1
HOW TO INVOKE THE DEBUG PROGRAM	9-1
DEBUG	9-1
DEBUGGING COMMANDS	9-4
COMMAND PARAMETERS	9-4
A (ASSEMBLE)	9-7
C (COMPARE)	9-10
D (DUMP)	9-11
E (ENTER)	9-14
F (FILL)	9-17
G (GO)	9-18
H (HEX)	9-21
I (INPUT)	9-23
L (LOAD)	9-24
M (MOVE)	9-27
N (NAME)	9-29
O (OUTPUT)	9-31
P (PROCEED)	9-32
Q (QUIT)	9-33
R (REGISTER)	9-34
S (SEARCH)	9-37

T (TRACE)	9-39
U (UNASSEMBLE)	9-40
W (WRITE)	9-42

A. ASCII DISPLAY AND KEYBOARD CODE TABLES

BASIC ASCII CODE (ISO 7-bit code)	A-1
EXTENDED ASCII CODE FOR THE DISPLAY (ISO 8-bit code)	A-5
NATIONAL VARIATIONS IN EXTENDED ASCII CODE (ISO 8-bit code)	A-8
EXTENDED KEYBOARD	A-14

B. ANSI ESCAPE SEQUENCES

INTRODUCTION	B-1
CURSOR MOVING FUNCTIONS	B-2
ERASING FUNCTIONS	B-4
GRAPHIC MODE FUNCTIONS	B-5
KEY REASSIGNMENT	B-7

C. CONFIGURING MS-DOS

INTRODUCTION	C-1
CONFIGURING COMMANDS	C-1
BREAK	C-1
BUFFERS	C-2
COUNTRY	C-3

DEVICE	C-5
DRIVPARM	C-6
FCBS	C-8
FILES	C-9
LASTDRIVE	C-10
SHELL	C-11
STACKS	C-12
 D. ERROR MESSAGES	
DEVICE AND DISK DRIVE ERRORS	D-1
ERROR MESSAGES IN ALPHABETICAL ORDER	D-3
 E. GLOSSARY OF TERMS	
GLOSSARY OF TERMS	E-1
 F. THE VIRTUAL DISK SYSTEM	
INTRODUCTION	F-1
INSTALLING VDISK.SYS	F-1
 G. INSTALLING NEW BLOCK DEVICES	
INTRODUCTION	G-1
INSTALLING DRIVER.SYS	G-1
 INDEX	

1. INTRODUCTION

ABOUT THIS CHAPTER

This chapter introduces some of the more commonly used features of MS-DOS, provides some information on disk handling and defines the notation convention used throughout this book.

CONTENTS

WHAT IS MS-DOS?	1-1	WRITE-PROTECTION	1-6
COMMAND LIBRARY	1-1	VIRTUAL DISK	1-8
FILE AND DIRECTORY HANDLING	1-1	NOTATION CONVENTIONS	1-9
PROGRAMMING TOOLS	1-1		
INTERNAL AND EXTERNAL COMMANDS	1-2		
BATCH PROCESSING	1-2		
THE AUTOEXEC.BAT FILE	1-2		
REAL-TIME CLOCK FUNCTIONS	1-2		
INTERFACE HANDLING	1-3		
DISKS	1-3		
DISKETTE HANDLING	1-5		
PURCHASING DISKETTES	1-6		
LABELING DISKETTES	1-6		

INTRODUCTION

WHAT IS MS-DOS?

The Microsoft Disk Operating System (MS-DOS) is a group of programs that controls the running and operation of your computer. It provides an interface between you and your computer.

Through MS-DOS you communicate with the Central Processing Unit (CPU), monitor, disk drives, printer, and other peripherals. The Disk Operating System enables you to manipulate program and data files stored on diskette or hard disk.

MS-DOS is supplied on your MS-DOS diskette.

The major features of MS-DOS are outlined in the sections that follow.

COMMAND LIBRARY

MS-DOS has a command library of over 40 commands that provide you with an environment suitable for handling files of information, developing programs, and running applications.

FILE AND DIRECTORY HANDLING

File handling commands not only allow you to copy and delete files, copy entire diskettes, display the contents of files, etc., but also to group files into directories at your convenience. Moreover, MS-DOS enables you to create directories within a directory, thereby creating a hierarchical directory structure. Refer to Chapter 3 for details.

PROGRAMMING TOOLS

A set of programming tools which enables you to write and develop programs. You can edit program files using the Video File Editor (see Chapter 6) or Line Editor EDLIN (see Chapter 7), link object files using the LINK utility (see Chapter 8), and debug executable files using the DEBUG utility (see Chapter 9). Macro Assembler, PASCAL, FORTRAN and other high level languages are separately available, to produce executable files. The GW-BASIC interpreter is available on your system disk for interpretive programming.

INTERNAL AND EXTERNAL COMMANDS

When MS-DOS is initialized some commands are loaded into memory and reside there. Other commands remain on disk. The former are known as internal commands, the latter are external commands and are loaded into memory and executed when required. Most of these external commands, after they are executed, are removed from memory, thereby optimizing the use of memory. However a few of these external commands, (GRAFTABL, GRAPHICS, PRINT, and SHARE), remain resident in memory after they have executed (see Chapter 5 for details of these commands).

BATCH PROCESSING

MS-DOS enables a commonly executed series of commands to be grouped into one file -a batch file -that can be executed simply by entering the file name. Refer to Chapter 4 for details.

THE AUTOEXEC.BAT FILE

The AUTOEXEC.BAT file is a special batch file which, if present, is executed automatically at system initialization. This is useful if your application requires a certain sequence of commands to be executed every time the system is initialized (see Chapter 4).

REAL-TIME CLOCK FUNCTIONS

When no AUTOEXEC.BAT file is present MS-DOS asks you the DATE and TIME.

MS-DOS has two commands that utilize the Real-Time Clock. These are:

- DATE which enables you to set the date
- TIME which enables you to set the time

These are important not only for programs that use time-dependent functions, but also because MS-DOS provides you with information about the time and date of creation or modification of your files.

INTRODUCTION

INTERFACE HANDLING

MS-DOS allows you to communicate with compatible devices (plotters, printers, modems, etc.) via the standard RS-232-C serial interface. You will need to set the protocol for the interface using the MODE command (see Chapter 5).

DISKS

Information is stored either on 3 1/2 in. or on 5 1/4 in. floppy disks or, if your system has, on hard disk. This manual will refer to the former as diskettes and the latter as the hard or fixed disk. The term "disk" will be used to mean either diskette or hard disk.

Drive letters (A,B,C through Z) are the means by which commands can identify a particular drive.

The drive letter of the first diskette drive in any system is A. The drive letter of any second diskette drive is B. The drive letter of the first hard disk is C. Drive letters D through Z are used for additional hard disks, disk partitions, virtual disks and dummy drives.

The capacity, of a disk drive determines the type of diskettes that can be used in it.

Diskettes can have a variety of capacities to hold data, as illustrated in the following table. See your Installation and Operations Guide to check what types of Diskette capacities your disk drive(s) can read and write.

Diskette Capacities

	Double Density 40 track (48 t.p.i.)		High Density 80 track (96 t.p.i. or 135 t.p.i.)		
	8 sector	9 sector	8 sector	9 sector	15 sector
Single Sided	160 KB	180 KB	320 KB	360 KB	-
Double Sided	320 KB	360 KB	640 KB	720 KB	1.2 MB

3 1/2 Inch Diskette Compatibility

These diskettes are 135 tracks per inch (t.p.i.) and can be formatted single or double sided with 80 tracks. Normally each track will be formatted with nine sectors, giving single sided disks a capacity of 360 KB and double sided disks a capacity of 720KB. A 3 1/2 inch disk drive with two heads can read and write single sided and double sided diskettes. A 3 1/2 inch disk drive with one head can only read and write single sided diskettes. Obviously you should not place 3 1/2 inch diskettes in a 5 1/4 inch drive, nor can you place 5 1/4 inch diskettes into a 3 1/2 inch drive.

5 1/4 Inch Diskette Compatibility

Standard formatting in Normal-Capacity drives is 40 tracks, 9 sectors per track. Formatting in High-Capacity drives is 80 tracks, 15 sectors per track. To format diskettes as 40 tracks, 9 sectors per track in High-Capacity drives use the /4 switch. However note that Normal-Density diskettes written to in High-Capacity drives cannot be reliably read in Normal-Capacity drives. To prevent accidental writing to Normal-Density diskettes in a High-Capacity drive, write protect the diskette.

The following figure shows 5 1/4 inch diskette compatibility in different drives:

INTRODUCTION

		DRIVE		
		Normal Capacity		High Capacity
		Single sided (160/180 KB)	Double sided (320/360 KB)	Double sided (1.2MB)
D 48 I tpi S K E T T 96 E tpi S	Single sided	Read/Write	Read/Write	Read
	Double sided	—	Read/Write	Read/Write*
	Double sided	—	—	Read/Write

* Once written the diskette cannot be reliably read in Normal Capacity Double Sided Drives.

Fig. 1-1 Diskette Type Compatibility in Different Capacity Drives

DISKETTE HANDLING

Although diskettes are generally durable, damage to diskettes will be minimized if you take the following precautions:

- Never bend 5 1/4 inch diskettes.
- Do not touch the exposed surface of the diskette or allow liquids, dust or cigarette ash to come into contact with it.
- Never expose your diskettes to strong magnetic fields, for example keep them away from telephones and tape recorders.
- Keep your diskettes away from direct sunshine, and store them in a cool place.

- Always keep a 5 1/4 inch diskette in its cardboard envelope when not in use
- Always file them in the diskette carton.
- Keep dust out of the diskette drives by keeping the drive covers closed when not in use.
- Do not attach anything to diskettes with paper clips or rubber bands.

PURCHASING DISKETTES

When purchasing your media make sure that the diskettes are Double Sided for Double Sided Disk Drives. High-Capacity 5 1/4 inch disk drives require 96 t.p.i. High-Density Diskettes. 3 1/2 inch disk drives require 135 t.p.i. Micro Floppy Diskettes.

LABELING DISKETTES

Every carton of diskettes contains a supply of self-adhesive labels for identifying diskettes. It is good practice to write all relevant details on the label before attaching it to the diskette. But if you do find it necessary to write on the label after sticking it to a 5 1/4 inch diskette, you should avoid using sharp pencils or ball-point pens as these may damage the surface of the diskette. In this case a felt-tipped pen is recommended.

WRITE-PROTECTION

To protect your data from being accidentally overwritten, you can apply write-protection to your diskettes.

5 1/4 Inch Diskette Write-Protection

For 5 1/4 inch diskettes a sheet of aluminized write-protect tabs is provided with every carton of diskettes. To apply write-protection, simply stick a tab over the write-protect notch cut into the side of the diskette. To remove write-protection, simply remove the tab. See the following figure which shows you the position of the write protect notch:

INTRODUCTION

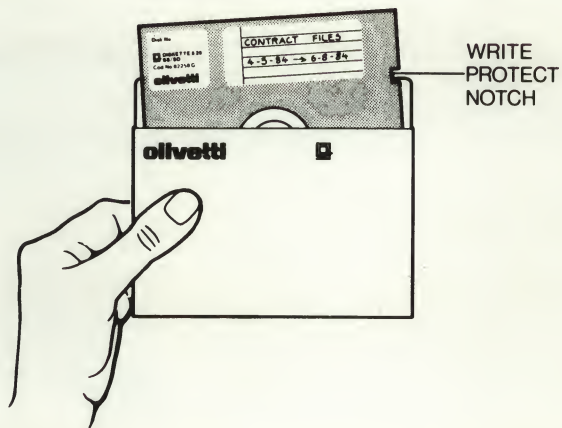


Fig. 1-2 The position of the Write Protect Notch on 5 1/4 Inch Diskettes

3 1/2 Inch Diskette Protection

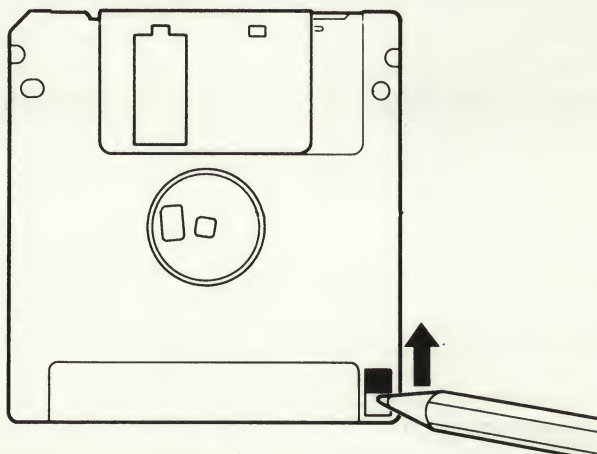


Fig. 1-3 Unprotected 3 1/2 Inch Diskette

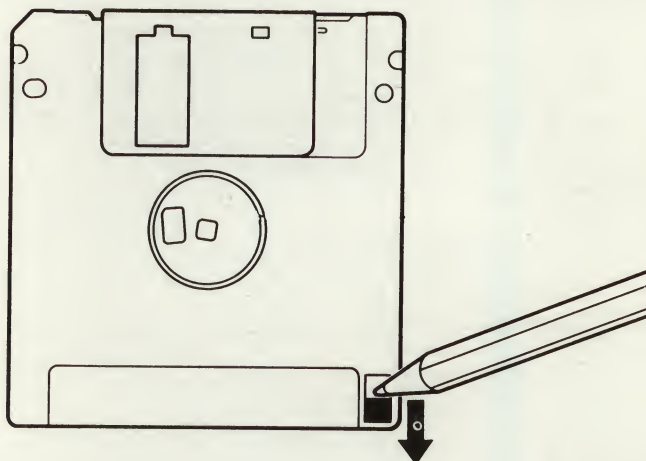


Fig. 1-4 Write Protected 3 1/2 Inch Diskette

On 3 1/2 inch diskettes there is a movable tag on the reverse side in the right hand corner (see the Figure "Unprotected 3 1/2 Inch Diskette"). The first figure shows the diskette with the tag up, this diskette is not write protected. Slide the tag down to the bottom of the slot, it clicks into place (see the Figure "Write Protected 3 1/2 inch Diskette"). Now if the computer tries to write to this diskette or to delete any files on the diskette, the result will be an error message:

Write protect error writing drive A
Abort, Retry, Ignore

If you really wish to write to the diskette, remove it from the drive, slide the tag up, until it clicks into place, replace it in the drive; then press **R** to retry. If you had the wrong disk in the drive do not change disks at this stage, instead press **A** to abort the operation; then exchange the diskettes.

VIRTUAL DISK

A virtual disk is part of main memory which emulates a backing store

INTRODUCTION

disk. The virtual disk appears to the operating system just like any other disk drive. For example, if a virtual disk is installed on a computer with two floppy disk drives "A:" and "B:", the virtual disk is drive "C:". The only difference between virtual disk and real disk is that when you turn your machine off, the information on virtual disk will be lost. So remember to COPY all files you want to keep from virtual disk to a real disk, before you turn you machine off. See Appendix F "The Virtual Disk System" for details on installing virtual disk.

NOTATION CONVENTIONS

The following notation conventions are used throughout this book:

- Uppercase, bold letters and words within a syntax line represent keywords that must be typed exactly as shown.

Example:

In the command line:

DISKCOPY [*sourcedrive:*] [*targetdrive:*]

DISKCOPY should be typed as shown.

Outside syntax lines, keywords are shown in uppercase but not in bold.

Note that uppercase letters and words are used simply as visual aids in this manual. Keywords may be typed in lowercase if desired.

- Lowercase italicized characters and words represent parameter names. They indicate that variable information is to be provided by the user.

Example:

In the command line:

DISKCOPY A: B:

both *sourcedrive* and *targetdrive* have been replaced by specific values, that is A and B.

- Hyphens may join lowercase letters or words to form a single parameter name.

Example:

In the command line:

R [register-name|F]

register-name is a single variable item that should be replaced by a single specific value, for example AX.

- A blank, a comma, a colon, or a semicolon may be used to separate the items in a line. In this manual the blank is usually shown in syntax lines.
- The symbols listed below are used to define the syntax of a line, but should not be typed in the actual line:

[] brackets

{ } braces

| vertical stroke ("or" sign)

... ellipsis

- Items contained by brackets ([]) are optional and so may be selected or not.

INTRODUCTION

Example:

The representation:

[*filespec*]

indicates that a *filespec* may be entered or omitted.

- Items enclosed by braces ({}) and separated by vertical strokes (|) are alternatives. You should select only one such alternative.

Example:

The representation:

{A|B|C}

indicates that either A or B or C should be selected.

- Items enclosed by brackets ([]) and separated by vertical strokes (|) are optional alternatives. You may choose one such alternative, or none at all.

Example:

The representation:

[A|B|C]

indicates that A or B or C may, but need not, be selected.

- An ellipsis indicates that the preceding item or group of items may be repeated more than once in succession.

Example:

The representation:

A [B]...

indicates that A can be typed alone or can be followed by one or more occurrences of:

B

- Letters and words in bold indicate MS-DOS messages that appear on your Personal Computer screen. For example:

**Insert new diskette for drive B:
and strike ENTER when ready**

- Letters and words shown in condensed bold indicate that you must press a specific key. For example the key whose inscription is CTRL is always referred to as **CTRL**.
- Commands need to be confirmed by pressing ↵ (the **ENTER** key), at the end of the command line.

2. MS-DOS CONTROL KEYS AND EDITING FUNCTION KEYS

ABOUT THIS CHAPTER

This chapter describes the keystroke combinations that are required to execute control functions such as Enter, carriage return, line feed, break, etc. It also indicates the keystrokes that perform special editing functions. For further details of the latter refer to Chapter 7.

CONTENTS


INTRODUCTION	2-1
MS-DOS CONTROL KEYS	2-1
EDITING FUNCTION KEYS	2-4

INTRODUCTION

Other than enabling you to enter text in the manner of a typewriter, your keyboard is capable, under MS-DOS, of allowing you to invoke other functions by using certain combinations of key-strokes. Such features fall into two groups: control keys and editing function keys.

MS-DOS CONTROL KEYS

MS-DOS control keys utilize a variety of key-stroke combinations. They are used to correct typing errors, abort command execution, etc. These functions are described in the following table. Note that if you are using an Olivetti extended function key Keyboard, the key combinations will have minor differences from those stated here (see your Installation and Operations Guide).

FUNCTION	KEY COMBINATION	MEANING
ENTER	 or CTRL M	The command line is terminated and command execution begins. Note that this key will be referred to as ENTER throughout this book.
Shift	↑ (SHIFT)	Activates the shifted value for a key (or gives a lower case value when CAPS LOCK is depressed).
Shift lock	CAPS LOCK	All alphabetic keys take upper case values. Press once and this key locks (toggles on). Press it again and it releases (toggles off).

FUNCTION	KEY COMBINATION	MEANING
Keypad lock	NUM LOCK	Activates the numeric keys on the right hand keypad. Press once and this key locks (toggles on). Press it again and it releases (toggles off). The keys on the numeric key pad, as a default generate their cursor function.
Backspace	← or CTRL H	The last character entered is removed from the command line and the character erased from the screen.
TAB	CTRL I or ␣ →	The cursor is moved to the next eight-character TAB position.
Line Feed	CTRL ENTER or CTRL J	Moves the cursor to the start of the next screen line where you can continue to enter the line you are typing.
Escape	ESC	Cancels the current line and moves the cursor to the next line of the screen. A backslash (\) is displayed at the end of the cancelled line.

FUNCTION	KEY COMBINATION	MEANING
Abort	CTRL BREAK or CTRL C	Aborts the current command. The command cannot be recommenced at the point of termination. If you wish to re-execute the command you must enter it again.
Suspend	CTRL NUM LOCK or CTRL S	Output to the screen is suspended. You can restart the display by pressing any other character key.
Echo Output	CTRL PRT SC or CTRL P	All output that is normally displayed on the screen is sent to the printer. Press the same keys again to terminate printer echo.
Print Screen	SHIFT PRT SC	Sends to the printer a copy of whatever is displayed on the screen. For graphic screens the MS-DOS command GRAPHICS has to be entered before pressing these keys (see Chapter 5).
System Reset	CTRL ALT DEL	Ends the current working session and reboots.

EDITING FUNCTION KEYS

These commands are entered by pressing a single key. Most of these commands use the function (F) keys.

When you press **ENTER** to execute a command, a copy of the command is kept in a special buffer called the source line. MS-DOS has a range of commands that you can use to enter a command line by copying all or part of the source line. For example if you wish to copy several files from drive B to drive A you might copy the first of those files by typing:

COPY B:MYFILE A:
then press **ENTER**

When you press **ENTER** this command would be written to the source line. If the next file you want to copy is called **YOURFILE**, then the character strings "COPY B:" and "FILE A:" can be copied from the source line using the function keys. The only part you need type in yourself is the string "YOUR".

The most common use of editing function keys, however, is in performing edit operations within a line of text when using the line editor **EDLIN**. A detailed description is therefore given in Chapter 7, but a brief description of each of the editing function keys is given in the following table:

FUNCTION	KEY COMBINATION	MEANING
BACKUP	←	Erases one character from command line.
COPY1	F1 or →	Copies one character from the source line to the command line.
COPYTO	F2 then type a <i>character</i>	Copies all characters, up to the first occurrence of <i>character</i> , from the source line to the command line.

MS-DOS CONTROL KEYS AND EDITING FUNCTION KEYS

FUNCTION	KEY COMBINATION	MEANING
COPYLINE	F3	Copies all remaining characters in the source line to the command line.
SKIP1	DEL	Skip over one character in the source line.
SKIPTO	F4 then type a <i>character</i>	Skip over characters in the source line up to <i>character</i> .
KILL	ESC	Terminates input and terminates the command line.
INSERT	INS	Enters/exits insert mode.
NEWTEMP	F5	Creates a new source line by copying the command line to the source line, but does not execute.

3. FILES AND DIRECTORIES

ABOUT THIS CHAPTER

This chapter describes how to manipulate files and directories, how you can access files in directories by means of paths and how to create and delete directories.

For further details of commands mentioned in this chapter refer to Chapter 5.

CONTENTS

HOW MS-DOS KEEPS TRACK OF YOUR FILES	3-1	PATHS AND EXTERNAL COMMANDS	3-10
HOW TO NAME YOUR FILES	3-1	HOW TO DISPLAY YOUR CURRENT DIRECTORY	3-11
WILD CARDS	3-2	HOW TO CREATE A DIRECTORY	3-13
THE ? WILD CARD	3-2	HOW TO CHANGE YOUR CURRENT DIRECTORY	3-13
THE * WILD CARD	3-3	HOW TO REMOVE A DIRECTORY	3-14
RESERVED DEVICE NAMES	3-4	HOW TO CHECK THE VALIDITY OF YOUR FILES	3-14
HOW TO COPY YOUR FILES	3-4		
HOW TO PROTECT YOUR FILES	3-5		
HOW TO EXAMINE THE CONTENTS OF A FILE	3-6		
DIRECTORIES	3-6		
FILE NAMES AND PATHS	3-8		
PATHNAMES	3-8		

FILES AND DIRECTORIES

HOW MS-DOS KEEPS TRACK OF YOUR FILES

The names of files are kept in directories on disk. These directories also contain information on the size of the files, their location on the disk, and the dates that they were created or updated. The directory you are working in is called your current directory.

An additional system area is called the File Allocation Table. It keeps track of the location of your files on the disk. It also allocates the free space on your disks so that you can create new files.

These two system areas, the directories and the File Allocation Table, enable MS-DOS to recognize and organize the files on your disks. The File Allocation Table is created on a disk when you format it with the MS-DOS FORMAT command. One empty directory is also created on it, known as the root directory.

HOW TO NAME YOUR FILES

A file name can comprise:

- One to eight characters (for legal characters see below). For example: NEWFILE.
- One to eight characters, followed by a period (.) and a one to three character file name extension. For example NEWFILE.EXE.

A file name may be made up of any of the following characters:

A-Z	0-9	\$	&	#	~
%	'	()	-	_
@	^	{	}	!	a-z

Alphabetic characters within the file name can be entered in upper or lower case, but MS-DOS will translate lower case letters into upper case.

How you specify a file depends on which directory of which disk drive it resides:

- If the file is in the current directory on the default drive you need only specify the file name, for example:

MYFILE

- If you are using tree-structured directories (see the section entitled "Directories" later in this chapter) and the file is on the default drive but not in the current directory, then you need to specify a path; for example:

\MIDIR\MYFILE

Path names are described later in this chapter in the section entitled "File Names and Paths".

- If the file is not on the default drive, you will need to specify the drive; for example:

B:MYFILE

WILD CARDS

Two special characters (called wild cards) can be used in file names and extensions: the asterisk (*) and the question mark (?). These special characters give you greater flexibility when using file names in MS-DOS commands.

THE ? WILD CARD

A question mark (?) in a file name or file name extension indicates that any character can occupy that position. A question mark(?) as the final character in a file name or file name extension indicates a character or no character can occupy that position. For example, the MS-DOS command:

DIR TEST?RUN.EX?

will list all entries in the current directory on the default drive that begin with TEST, have any next character, end with the letters RUN, and

FILES AND DIRECTORIES

have a file name extension of EX followed by any character or no character. Here are some examples of files that might be listed by the above DIR command:

```
TEST0RUN.EX
TEST1RUN.EXE
TEST2RUN.EXE
TEST6RUN.EXE
```

THE * WILD CARD

An asterisk (*) in a file name indicates that any valid character or sequence of valid characters can occupy that position in the file name. Any characters after the * are ignored by the system. For example:

```
DIR TEST*.EXE
```

will list all entries in the current directory on the default drive with file names that begin with the characters TEST and have an extension of EXE. Here are some examples of files that might be listed by the above DIR command:

```
TEST1RUN.EXE
TEST2RUN.EXE
TEST6RUN.EXE
TESTALL.EXE
```

The wild card designation *.* refers to all files in the current directory. Note that this can be very powerful when used in MS-DOS commands. For example, the command:

```
COPY A:*. * C:
```

copies all files on the current directory of the A: drive, regardless of file name, to the current directory of the C: drive.

Examples

To list all files named NEWFILE with any extension in the current directory on drive A, simply enter:

```
DIR A:NEWFILE.*
```


To list all file names that have less than or equal to six-characters plus an extension of .TXT in the current directory of the diskette in B, enter:

DIR B:?????.TXT

RESERVED DEVICE NAMES

MS-DOS treats device names specially, and certain words are reserved for the names of these devices. These names cannot be used as file names or extensions. Reserved names are as follows:

AUX: or COM1: Used when referring to input from or output to the built-in communications port.

COM2: Used when referring to input from or output to a second communications port.

CON: Used when referring to keyboard input or screen output.

PRN: or LPT1: Used when referring to the first parallel printer.

LPT2: or LPT3: Used when referring to the second and third parallel printers.

NUL: Used when you do not want to create a particular file, but the command requires an input or output file name.

The colon following the reserved device name is optional.

You can use a device name instead of a file name. Using "Redirection" (see Chapter 4) data can then be input from a device or output to a device, instead of a file.

HOW TO COPY YOUR FILES

You often need more than one copy of a file. The COPY command allows you to copy one or more files to another disk, to another file on the same disk or to another directory on the same disk (see the section entitled "File Names and Paths"). You can also give the copy a different name if you specify the new name in the COPY command.

FILES AND DIRECTORIES

You cannot make a copy of a file in the same directory unless you specify a different file name for the new copy.

For example,

```
COPY A:MYFILE.TXT B:MYFILE.TXT
```

will copy the file MYFILE.TXT on the diskette in drive A to a file named MYFILE.TXT on the diskette in drive B.

If you want to duplicate the file named MYFILE.TXT on the same disk, enter:

```
COPY A:MYFILE.TXT A:NEWNAME.TXT
```

You now have two copies of your file on drive A one named MYFILE.TXT and the other named NEWNAME.TXT.

You can also use the COPY command along with the wild card feature, to copy groups of files from one disk to another, or to copy all files from one disk to another.

For example, the command

```
COPY A:*.EXE C:
```

copies all the .EXE files from the current directory of the A: drive, to the current directory of the C: drive.

HOW TO PROTECT YOUR FILES

MS-DOS is a powerful and useful environment for processing your personal and business information. As with any information system, inadvertent errors may occur and information may be misused. If you are processing information that cannot be replaced or requires a high level of security, you should take steps to ensure that your data and programs are protected from accidental or unauthorized use, modification, or destruction. Simple measures you can take, such as removing your disks when they are not in use, keeping back-up copies of valuable information, write-protection, and installing your equipment in a safe place, can help you maintain the integrity of the information in your files.

HOW TO EXAMINE THE CONTENTS OF A FILE

There are two commands that enable you to examine a file. These are:

- The TYPE command that displays the contents of a text file.
- The HEXDUMP file that displays a file in hexadecimal.

If you want to print a file you can do this by means of the ">" I/O redirection symbol. For example enter:

```
TYPE myfile > PRN
```

will send the contents of myfile to the printer instead of the screen. For details about I/O redirection refer to Chapter 4.

DIRECTORIES

The names of your files are kept in directories on each disk. Each directory also contains information on the size of the files and the dates that they were created or last updated.

If your computer is used by several people, or if you are working on several different projects, the number of files in the directory can become large and unwieldy. Moreover, this will certainly be the case if you are using a hard disk as it is capable of storing a vast number of files. You may want your own files kept separate from those of a colleague: or, you may want to organize your programs into categories that are convenient to you.

MS-DOS allows you to organize the files on your disks into directories. Directories are a way of dividing your files into convenient groups. For example, you may want all your accounting programs in one directory and text files in another. Any directory can contain another directory, this is a sub-directory. This method of organizing your files is called a hierarchical directory structure.

A hierarchical directory structure can be thought of as a "tree" structure: directories are branches of the tree and files are the leaves, except that the "tree" grows downward; that is, the "root" is at the top. The root is the first level in the directory structure. It is the directory

FILES AND DIRECTORIES

that is automatically created when you format a disk and start putting files in it. Any directory can contain files as well as sub-directories.

The tree structure grows as you create new directories for groups of files. Within each new directory, files can be added, or new subdirectories can be created.

It is possible for you to "travel" around this tree; for instance, you can find any file in the system by starting at the root and traveling down any of the branches to the desired file. Conversely, you can start where you are within the file system and travel towards the root.

The following figure illustrates a typical hierarchical directory structure:

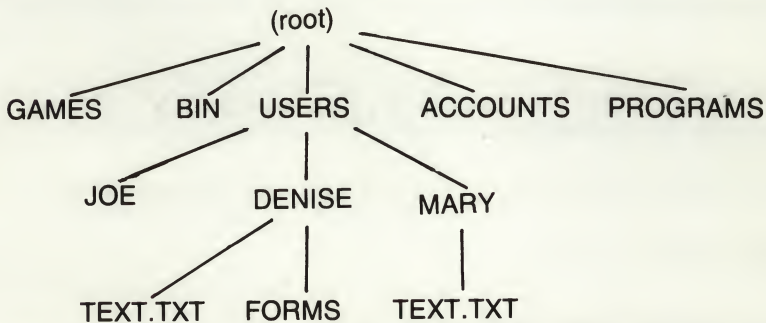


Fig. 3-1 A Sample Hierarchical Directory Structure

The root directory is the first level in the directory structure. You can create subdirectories from the root by using the MKDIR command. The root directory may also contain files.

In this example, five subdirectories of root have been created. These include:

- A directory of games, named GAMES.
- A directory of all external commands, named BIN.
- A USERS directory containing a separate subdirectory for each user of the system.

- A directory containing accounting information, named ACCOUNTS.
- A directory of programs, named PROGRAMS.

Joe, Denise and Mary each have their own directories which are sub-directories of the USERS directory. Denise has a subdirectory under the \USERS\DENISE directory named FORMS. Denise and Mary have files in their directories, each named TEXT.TXT. Notice that Mary's text file is unrelated to Denise's.

This organization of files and directories is not important if you only work with files in your own directory; but if you work with someone else or on several projects at one time, the hierarchical directory structure becomes extremely useful. For example, you could get a list of the files in Denise's FORMS directory by entering:

```
DIR \USERS\DENISE\FORMS
```

Note that the back-slash (\) is used to separate directories from other directories and files. The first back-slash in a directory sequence represents the root directory.

To find out what files Mary has in her directory, you could enter:

```
DIR \USERS\MARY
```

FILE NAMES AND PATHS

When you use hierarchical directories, you must tell MS-DOS where the files are located in the directory structure. Both Mary and Denise, for example, have files named TEXT.TXT. Each will have to tell MS-DOS in which directory her file resides if she wants to access it. This is done by giving MS-DOS a pathname to the file.

PATHNAMES

A pathname is a sequence of one or more directory names followed, optionally, by a file name, each separated from the previous one by a back-slash (\).

FILES AND DIRECTORIES

The syntax of a pathname is:

`[drive:][\][[directory][\directory]...\]filename`

or

`[drive:][\][[directory][\directory]...`

If a pathname begins with a backslash, MS-DOS searches for the file beginning at the root (or top) of the tree. Otherwise, MS-DOS begins at the user's current directory and searches downward from there. For example, the path of Denise's TEXT.TXT file is \USERS\DENISE\TEXT.TXT. That is, if you are in directory JOE and you want to access the file named TEXT.TXT in directory DENISE you would use the pathname:

\USERS\DENISE\TEXT.TXT

If you were in directory USERS, however, you could access the same file using the pathname:

DENISE\TEXT.TXT

However, if you are already in directory DENISE you can simply use the file name:

TEXT.TXT

If you are in directory MARY and you wish to make a copy of Mary's file TEXT.TXT in Denise's directory, but rename the copy COPY.TXT, you would enter:

COPY TEXT.TXT \USERS\DENISE\COPY.TXT

MS-DOS provides special shorthand notations for the current directory and the parent directory (one level up) of the current directory:

- . MS-DOS uses this shorthand notation to indicate the name of the current directory in all hierarchical directory listings. MS-DOS automatically creates this entry when a directory is made
- .. The shorthand name of the current directory's parent directory. If you enter:

DIR ..

then MS-DOS will list the files in the parent directory of your current directory.
If you enter:

DIR ..\..

then MS-DOS will list the files in the parent's parent directory.

(Note that the maximum length of a pathname is 63 characters.)

PATHS AND EXTERNAL COMMANDS

External commands reside on disk as program files. They must be read from the disk before they execute.

When you are working with more than one directory, it is convenient to put all MS-DOS external commands into a separate directory so they do not clutter your other directories. When you issue an external command to MS-DOS, MS-DOS immediately checks your current directory to find that command. You must tell MS-DOS in which directory these external commands reside. This is done with the PATH command.

For example, if your current directory is \BIN\PROG, and all MS-DOS external commands are in \BIN\COMMANDS, you must tell MS-DOS to choose the \BIN\COMMANDS path to find an external command. Enter the command:

PATH \BIN\COMMANDS

FILES AND DIRECTORIES

This tells MS-DOS to search first your current directory and then the \BIN\COMMANDS directory for all commands. You only have to specify this path once to MS-DOS during your working session. The PATH command can be in AUTOEXEC.BAT. If you want to know what the current path is, enter the PATH command without a parameter and the current value of PATH will be displayed.

You may also specify several paths. Refer to the PATH command in Chapter 5 for more information.

HOW TO DISPLAY YOUR CURRENT DIRECTORY

You can find out the name of the directory you are in by issuing the MS-DOS command CD or CHDIR (Change Directory) with no parameter. For example, if your current directory is \USERS\DENISE, when you enter:

CD

you will see:

C:\USERS\DENISE

This is the complete path of your current directory and comprises your current drive designation plus the current directory (\USERS\DENISE). If you now want to see what is in the \USERS\DENISE directory, you can issue the MS-DOS command DIR. The following is an example of the display you might receive for the DIR command for a subdirectory:

Volume in drive C is HARD DISK

Directory of C:\USERS\DENISE

.	<DIR>	8-09-82	10:09a
..	<DIR>	8-09-82	10:09a
FORMS	<DIR>	8-09-82	10:09a
TEXT	TXT	5243	8-04-82 9:30a
4 File(s) 8376320 bytes free			

Fig. 3-2 Subdirectory Example

Note that MS-DOS lists files and directories in this output. As you can see, DENISE has another directory in this tree structure named FORMS. The '.' indicates the current directory \USERS\DENISE, and the '..' is the shorthand notation for the parent directory USERS. TEXT.TXT is a file in the \USERS\DENISE directory. All of these directories and files reside on the disk in drive C.

Because files and directories are listed together (see previous display), MS-DOS does not allow you to give a subdirectory the same name as a file in that directory. For example, consider again the path USERS\DENISE\FORMS. FORMS is a subdirectory of DENISE, therefore you cannot create a file named FORMS in directory DENISE.

FILES AND DIRECTORIES

HOW TO CREATE A DIRECTORY

To create a subdirectory in your current directory, use the MD or MKDIR (Make Directory) command. For example, if you are in directory DENISE and you want to create a new directory named NEWDIR within your current directory, simply type:

```
MD NEWDIR  
then press ENTER
```

and a new directory will exist in your tree structure under your current directory.

You can also create directories anywhere in the tree structure by specifying MD and then a path. For example, if your current directory is DENISE and you want to create a directory named SPECS in directory MARY you would type:

```
MD \USERS\MARY\SPECS  
then press ENTER
```

MS-DOS will automatically create the . and .. entries in the new directory.

To create text files in the new directory, use either the Video File Editor (see Chapter 6) or the MS-DOS line editor EDLIN (see Chapter 7).

HOW TO CHANGE YOUR CURRENT DIRECTORY

To change your current directory to another directory simply issue the CD or CHDIR (Change Directory) command and supply a path. For example type:

```
CD \USERS  
then press ENTER
```

to change the current directory to \USERS. You can specify any path after the command to "travel" to different branches and leaves of the directory tree. The command "CHDIR .." will always put you in the parent directory of your current directory (unless you are in root).

HOW TO REMOVE A DIRECTORY

To remove a directory from the tree structure, use the MS-DOS RD or RMDIR (Remove Directory) command. For example, to remove the directory NEWDIR from the current directory, type:

```
RD NEWDIR  
then press ENTER
```

Note that the directory NEWDIR must be empty except for the . and .. entries before it can be removed; this will prevent you from accidentally deleting files and directories. You can remove any directory by specifying its path. To remove the \USERS\JOE directory, make sure that it has only the . and .. entries, then type:

```
RD \USERS\JOE  
then press ENTER
```

If the directory is not empty an error is reported. To remove all the files in a directory (except for the . and .. entries), enter DEL and then the path of the directory. For example, to delete all files in the \USERS\DENISE directory, type:

```
DEL \USERS\DENISE  
then press ENTER
```

and MS-DOS will display:

Are you sure? (Y/N)

Press **Y** and the files will be deleted. You cannot delete the . and .. entries (except by removing the directory).

HOW TO CHECK THE VALIDITY OF YOUR FILES

The CHKDSK command is used to check your disks for consistency and errors. The CHKDSK command analyzes the directories and the File Allocation Table on the disk that you specify. It then produces a status report of any inconsistencies, such as files which have a non-zero size in their directory but really have no data in them.

FILES AND DIRECTORIES

To check the disk in drive A type:

CHKDSK A:
then press **ENTER**

MS-DOS will display a status report and any errors that it has found. Moreover, if you include the flag switch **/V** in the command line:

CHKDSK A: /V

then the CHKDSK command displays messages as it runs and also shows the hidden files. You should run the CHKDSK command occasionally for each disk to ensure the integrity of your files.

4. ENTERING AND USING MS-DOS COMMANDS

ABOUT THIS CHAPTER

This chapter defines the syntax for a command, explains how commands can be grouped into batch files, and how the output from a command can be redirected to some device other than the VDU. It also describes the concept of "piping", whereby the output from one command becomes the input to another.

For further details of commands mentioned in this chapter refer to Chapter 5.

CONTENTS

COMMAND SYNTAX	4-1	REDIRECTING YOUR OUTPUT	4-15
PARAMETERS	4-1	REDIRECTING YOUR INPUT	4-16
INTERNAL AND EXTERNAL COMMANDS	4-6	FILTERS	4-16
EXTERNAL COMMAND SYNTAX	4-8	COMMAND PIPING	4-17
INFORMATION COMMON TO ALL MS-DOS COMMANDS	4-8		
BATCH PROCESSING	4-10		
HOW TO CREATE AND EXECUTE A BATCH FILE	4-10		
THE AUTOEXEC.BAT FILE	4-13		
BATCH FILES WITH REPLACEABLE PARAMETERS	4-14		
INPUT AND OUTPUT	4-15		

ENTERING AND USING MS-DOS COMMANDS

COMMAND SYNTAX

The general command format is defined as follows:

KEYWORD [*parameter*] ...

Where

SYNTAX ELEMENT	MEANING
KEYWORD	A one to eight character mnemonic that specifies the command to be executed. It must exclude any file name extension.
<i>parameter</i>	A parameter to the command defining the command action. The number of parameters depends on the command executed.

Remarks

If the KEYWORD is the name of an executable file, it may not reside on the default drive and directory. In this case the general command format can be extended (see the section on "External Command Syntax").

PARAMETERS

Parameters are user-selected strings of alphabetic characters and of integers which can be optional. They are recognized by their position in the command line. The types of parameter are described in the following table:

PARAMETER TYPE	MEANING																		
<i>filename</i> or <i>directory</i>	<p>Either a one to eight character string or a one to eight character string followed by a period (.) and a one to three character extension. It must be made up from the following characters:</p> <table><tr><td>A-Z</td><td>0-9</td><td>\$</td><td>&</td><td>#</td><td>~</td></tr><tr><td>%</td><td>'</td><td>(</td><td>)</td><td>-</td><td>_</td></tr><tr><td>@</td><td>^</td><td>{</td><td>}</td><td>!</td><td>a-z</td></tr></table> <p>Note: Lower-case letters are converted into upper case.</p> <p>For example:</p> <p>NEWFILE</p> <p>NEWFILE.TXT</p>	A-Z	0-9	\$	&	#	~	%	'	()	-	_	@	^	{	}	!	a-z
A-Z	0-9	\$	&	#	~														
%	'	()	-	_														
@	^	{	}	!	a-z														
<i>filespec</i>	<p><i>[drive:]filename</i></p> <p>A file specifier (<i>filespec</i>) can be a file name with or without a drive specifier (<i>drive</i>). For example:</p> <p>NEWFILE.TXT</p> <p>B:NEWFILE.TXT</p>																		
<i>pathname</i>	<p><i>[drive:][\][[directory][\directory]...\]filename</i></p> <p>or</p> <p><i>[drive:][\][[directory][\directory]...</i></p>																		

ENTERING AND USING MS-DOS COMMANDS

PARAMETER TYPE	MEANING
	<p>a pathname may comprise:</p> <ul style="list-style-type: none">• a drive specifier followed by a colon, specifying the current directory of the specified drive; for example: C:• one or more directory names separated by backslashes specifying a directory path relative to the current directory of the default drive; for example: MARY\PROGRAMS\PASCAL• a drive specifier followed by one or more directory names separated by backslashes, specifying a directory path starting from the current directory of the specified drive; for example: C:MARY\PROGRAMS\PASCAL• a backslash, specifying the root directory of the default drive: \• a drive specifier followed by a colon and a backslash specifying the root directory of the specified drive; for example: C:\• a backslash followed by one or more directory names separated, by backslashes specifying a directory path starting from the root directory of the default drive; for example: \USER\MARY\PROGRAMS\PASCAL

PARAMETER TYPE	MEANING
	<ul style="list-style-type: none"> • a drive specifier followed by a colon, a backslash and one or more directory names separated by backslashes, specifying a directory path starting from the root directory of the specified drive; for example: C:\USER\MARY\PROGRAMS\PASCAL • any of the above definitions followed by a file name • a file name
<i>drive</i>	A single letter specifying a diskette drive or hard disk.
<i>switch</i>	<p>An option which controls the way the command executes. It often takes the form of a single letter preceded by a slash. For example:</p> <p>/P</p>
<i>argument</i>	<p>Provides more information to an MS-DOS command. For example:</p> <p>ON or OFF</p>

ENTERING AND USING MS-DOS COMMANDS

PARAMETER TYPE	MEANING																		
<i>volume label</i>	<p>One to eleven character string. Spaces may be included but not tabs. A volume label must be made up from the following characters:</p> <table><tr><td>A-Z</td><td>0-9</td><td>\$</td><td>&</td><td>#</td><td>~</td></tr><tr><td>%</td><td>'</td><td>(</td><td>)</td><td>-</td><td>—</td></tr><tr><td>@</td><td>{</td><td>}</td><td>!</td><td>a-z</td><td></td></tr></table> <p>Note: Lower case letters are converted to uppercase. For example:</p> <p>Hard Disk -> HARD DISK SYSTEM DISK</p>	A-Z	0-9	\$	&	#	~	%	'	()	-	—	@	{	}	!	a-z	
A-Z	0-9	\$	&	#	~														
%	'	()	-	—														
@	{	}	!	a-z															

Nil Parameters

Some parameters are optional and take default values if they are not specified in the command line. For example, if you enter:

`MODE COM1:11`

then the last three parameters of the `MODE COM` command, which takes four parameters in all, take default values.

If default parameters are required in the middle of a command line commas must be entered separating omitted parameter. For example, if you enter:

`MODE COM1:11,,1`

then the omitted second and third parameters take default values.

INTERNAL AND EXTERNAL COMMANDS

There are two types of MS-DOS command:

- Internal commands
- External commands

Internal commands are the simplest, most commonly used commands. You cannot see these commands when you do a directory listing on your MS-DOS disk; they are part of the command processor. Therefore they reside in memory whenever MS-DOS is loaded. When you enter these commands, they execute immediately. This class of command comprises:

BREAK	DEL (ERASE)	MKDIR (MD)	SET
CHDIR (CD)	DIR	PATH	SHIFT
CLS	ECHO	PAUSE	TIME
COPY	EXIT	PROMPT	TYPE
CTTY	FOR	REM	VER
DATE	GOTO	REN (RENAME)	VERIFY
	IF	RMDIR (RD)	VOL

External commands reside on disk as program files. Any file name with a file extension of .COM, .EXE, or .BAT is considered to be an external command. They must be read from disk and loaded into memory before they can execute. With most of the external commands, following execution they are removed from memory. However a few of these external commands (GRAFTABL, GRAPHICS, PRINT and SHARE) remain resident in memory after they have executed.

ENTERING AND USING MS-DOS COMMANDS

When you enter an external command, do not include its file name extension. External commands include:

ASSIGN	FC	MORE
ATTRIB	FDISK	PRINT
BACKUP	FIND	RECOVER
CHKDSK	FORMAT	REPLACE
COMMAND	GRAFTABL	RESTORE
COMP	GRAPHICS	SELECT
DEBUG	GW BASIC	SHARE
DISKCOMP	HEXDUMP	SORT
DISKCOPY	JOIN	SUBST
EDIT	LABEL	SYS
EDLIN	LINK	TREE
EXE2BIN	MODE	XCOPY

So for example invoking `FORMAT` executes the command file `FORMAT.COM` and invoking `ATTRIB` executes the executable file `ATTRIB.EXE`. `.EXE` files have to be located in memory when they are loaded. Some `.EXE` files can be converted to `.COM` files using the MS-DOS utility `EXE2BIN`. `.COM` files are in memory image format and always load starting at location `100H` in a memory segment, therefore `.COM` format is more compact and loads faster. Because all external commands reside on disk, you can create commands and add them to the system by writing programs in assembler or high level languages and compiling them. MS Compilers and the assembler `MASM` produce object code (`.OBJ`) files. These `.OBJ` files have to be linked, using the linker `LINK` (see Chapter 8). The `LINK` produces `.EXE` (executable) files. If the `.EXE` cannot be converted to `.COM` files the following error message appears:

File cannot be converted

Refer to chapter 5 for more details on `EXE2BIN`.

When you specify an external command simply as `KEYWORD`, MS-DOS first looks in the default directory of the default drive. It then searches the paths set in the `PATH` variable of the environment. If the `KEYWORD` command file is not found, it cannot be executed and the following error message appears:

Bad command or filename

For external commands the general command format can be extended by preceding the KEYWORD with the Drive where the command file resides and/or the path leading to its directory.

EXTERNAL COMMAND SYNTAX

The general format of external commands is therefore defined as follows:

[*drive:*][*path*]KEYWORD[*parameter*]...

PARAMETER TYPE	MEANING
<i>drive:</i>	A one character drive specifier followed by a colon, specifying the drive where the KEYWORD is to be found.
<i>path</i>	{[<i>\</i>] <i>directory</i> [<i>\directory</i>]... <i>\</i> } If the path consists of the root directory, only one backslash should be used, for example: C:\

INFORMATION COMMON TO ALL MS-DOS COMMANDS

The following information applies to all MS-DOS commands:

- Commands are usually followed by one or more parameters.
- Commands and parameters may be entered in upper case or lower case, or a combination of both. MS-DOS will convert all lower case letters to upper case.


ENTERING AND USING MS-DOS COMMANDS

- Commands and parameters must be separated by delimiters. A space is usually used; for example:

`COPY A:MYFILE B:YOURFILE`

You can also use the comma (,), semicolon (;) or the equal sign (=) as delimiters in MS-DOS commands.

For clarity, this manual will use a space as the delimiter

- When you are instructed to "Press any key", you can press any alphabetic (A-Z) or numeric (1-9) key.
- You must include the file name extension when referring to a file that already has one.
- You can abort commands that perform input/output by pressing **CTRL C** or **CTRL BREAK**.
- Commands take effect only after you have pressed  **ENTER**.
- Wild cards (global file name characters) and device names (for example, PRN or CON) are not allowed in the names of any commands.
- When commands produce a large amount of output on the screen, the display will automatically scroll to the next screen. You can press **CTRL S** or **CTRL NUMLOCK** to suspend the display. Press any key to resume the display on the screen.
- MS-DOS control keys and editing function keys can be used when entering commands. Refer to Chapter 2 for a description of these keys.
- The default prompt from the command processor is the default drive designation plus a greater-than sign; for example, `A>`. You can change this prompt using the **PROMPT** command (see Chapter 5 for details).
- Disk drives will be referred to as source drive and target drive. A source drive is the drive you will be transferring information from. A target drive is the drive you will be transferring information to.

BATCH PROCESSING

Often you may find yourself entering the same sequence of commands over and over to perform some common task. With MS-DOS, you can put the command sequence into a special file called a batch file, and execute the entire sequence simply by entering the name of the batch file. "Batches" of your commands in such files are processed as if they were entered at the keyboard. Each batch file must be named with the .BAT extension, and is executed by entering the file name without its extension.

HOW TO CREATE AND EXECUTE A BATCH FILE

You can create a batch file by using the Video File Editor, the Line Editor (EDLIN) or the COPY command.

The MS-DOS command library contains a sub-set of batch processing commands. Among the more commonly used are REM and PAUSE. REM permits you to include remarks and comments in your batch files without these remarks being executed as commands. PAUSE prompts you with an optional message and permits you to either continue or abort the batch process at a given point.

Batch processing is useful if you want to execute several MS-DOS commands with one batch command, such as when you format and check a new diskette. For example, a batch file for this purpose might look like this:

```
REM This is a file to check new diskettes  
REM It is named NEWDISK.BAT  
PAUSE Insert new diskette in drive B  
FORMAT B:  
CHKDSK B:
```

Place an unprotected working copy of your system diskette in drive A.

Make sure that you have the prompt A> and proceed as follows to create the example batch file:

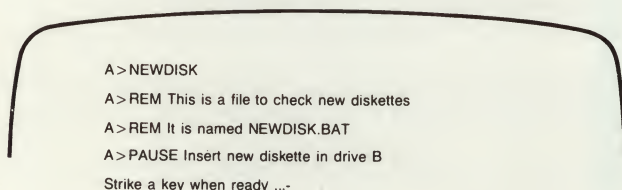
ENTERING AND USING MS-DOS COMMANDS

STEP	IF you enter...	THEN...
1	COPY CON: NEWDISK.BAT	the computer awaits input from the keyboard (CON:)
2	REM This is a file to check new diskettes	"REM This is a file to check new diskettes" is entered on the first line
3	REM It is named NEWDISK.BAT	"REM It is named NEWDISK .BAT" is entered on the second line
4	PAUSE Insert new diskette in drive B	"PAUSE Insert new diskette in drive B" is entered on the third line
5	FORMAT B:	"FORMAT B:" is entered on the fourth line
6	CHKDSK B:	"CHKDSK B:" is entered on the fifth line
7	CTRL Z	type CTRL Z the end-of-file character is entered on the sixth line
8	ENTER	press ENTER the file creation is complete and the message: 1 File(s) copied appears on the screen. The file NEWDISK.BAT is created on the system diskette.

To execute this batch file, simply enter the file name without the extension:

NEWDISK

The result is the same as if each of the lines in the batch file were entered at the terminal as individual commands. That is, the first three commands are executed successively and the following messages are displayed on the screen:



```
A>NEWDISK
A>REM This is a file to check new diskettes
A>REM It is named NEWDISK.BAT
A>PAUSE Insert new diskette in drive B
Strike a key when ready ...-
```

Fig. 4-1 Sample Batch File Display

After striking a key, the diskette in drive B is formatted. You will then be asked if you wish to format another diskette. Following a negative reply, the diskette will be checked.

Remarks

1. Only the file name should be entered to execute the batch file. Do not enter the file name extension.
2. Do not name batch files with internal command names.
3. If you name batch files with external command names, will be executed in preference to the .BAT file, .EXE or .COM file.

ENTERING AND USING MS-DOS COMMANDS

4. If you press **CTRL C** or **CTRL BREAK** while in batch mode, this prompt appears:

terminate batch job (Y/N)?

If you press **Y**, the remainder of the commands in the batch file are ignored and the system prompt appears.

If you press **N**, only the current command is terminated and batch processing continues with the next command in the file.

5. If you remove the diskette containing a batch file being executed, MS-DOS prompts you to insert it again so the next command can be read.
6. The last command in a batch file may be the name of another batch file. This allows you to call one batch file from another, when the first is finished. However there is no return to the calling batch file.
7. Input and Output can be redirected (the "**<**", "**>**", "**> >**") symbols. See later in this chapter for more information.

THE AUTOEXEC.BAT FILE

An **AUTOEXEC.BAT** file is a batch file that allows you to automatically execute programs when you start MS-DOS. Automatic Program Execution is useful when you want to run a specific package (for example, Multiplan) under MS-DOS, and when you want MS-DOS to execute a batch program automatically each time you start the system.

When you start MS-DOS, the command processor searches the MS-DOS system diskette for a file named **AUTOEXEC.BAT**. The **AUTOEXEC.BAT** file is a batch file that is automatically executed each time you start the system.

The **AUTOEXEC.BAT** file is created in exactly the same way as any other batch file. It must, however, reside in the root directory of the MS-DOS system disk.

Example

If your AUTOEXEC.BAT file contains the following:

```
DATE  
TIME  
GWBasic
```

then on initializing your system the date and time prompts will appear and the system will automatically enter GWBasic.

BATCH FILES WITH REPLACEABLE PARAMETERS

You may want commands within a batch file to have replaceable parameters. For example, if your batch file contains a COPY command, you may wish to supply a different parameter to the COPY command each time you run the batch file. You can do this by specifying dummy parameters to the commands within the batch file. These parameters, named %0 to %9, can be replaced by values supplied when the batch file executes. For example, you may have created the following batch file named "MYFILE.BAT":

```
COPY %1.MAC %2.MAC  
TYPE %2.TXT  
TYPE %0.BAT
```

To execute this file you must enter the file name without extension, which is the value for parameter %0, followed by the replacement values for %1 and %2. For example, if you type:

```
MYFILE A:PROG1 B:PROG2  
then press the ENTER key
```

then:

- %0 is replaced by "MYFILE"
- %1 is replaced by "A:PROG1"
- %2 is replaced by "B:PROG2"

ENTERING AND USING MS-DOS COMMANDS

The effect is to execute the following sequence:

COPY A:PROG1.MAC B:PROG2.MAC

TYPE B:PROG2.TXT

TYPE MYFILE.BAT

Remarks

1. Up to 10 dummy parameters (%0-%9) can be specified in this way. Refer to the SHIFT command if you wish to specify more than 10 dummy parameters.
2. If you use the percent sign as part of a file name within a batch file, you must enter it twice. For example, to specify the file ABC%.EXE, you must enter it as ABC%%.EXE in the batch file.

INPUT AND OUTPUT

MS-DOS normally assumes that input comes from the keyboard and that output goes to the screen. However, the flow of command input and output can be redirected. Input can come from a file rather than the keyboard, and output can go to a file or to a printer instead of to the screen. In addition, "pipes" can be created that allow output from one command to become the input to another. Redirection and pipes are discussed in the next sections.

REDIRECTING YOUR OUTPUT

Most commands produce output that is sent to the screen. You can send this information to a file by using a greater-than sign (>) in your command. For example, the command:

DIR

displays a directory listing of the current directory on the screen. The same command can send this output to a file named MYFILES instead of the screen by designating the output file in the command line:

DIR >MYFILES

If the file MYFILES does not already exist, MS-DOS creates it and stores your directory listing in it. If MYFILES already exists, MS-DOS overwrites what is in the file with the new data.

Two greater-than signs (> >) can be used to tell MS-DOS to append the output of the command (such as a directory listing) to the end of a specified file. For example, the command:

```
DIR >>MYFILES
```

appends your directory listing to a currently existing file named MYFILES. If MYFILES does not exist, it is created.

REDIRECTING YOUR INPUT

It is often useful to have input for a command come from a file rather than from the keyboard. This is possible in MS-DOS by using a less-than sign (<) in your command. For example, the command:

```
SORT <NAMES >LIST1
```

sorts the file NAMES and sends the sorted output to a file named LIST1.

FILTERS

A filter is a command that reads your input, transforms it in some way, and then sends the output, usually, to the screen or to a file. In this way, the data is said to have been "filtered" by the program. Since filters can be put together in many different ways, a few filters can take the place of a large number of specific commands.

MS-DOS filters include FIND, MORE, and SORT. Their functions are described below:

FIND Searches for a particular string of text in a file.

MORE Takes standard output and displays it, one screen at a time.

SORT Sorts text.

Refer to Chapter 5 for details of these commands.

You can see how these filters are used in the next section.

COMMAND PIPING

If you want to give more than one command to the system at a time, you can "pipe" commands to MS-DOS. For example, you may occasionally need to have the output of one program sent as the input to another program. A typical case would be a program that produces output in columns. You might want to have this columnar output sorted.

Piping is done by separating commands with the pipe separator, which is the vertical bar symbol (|). For example, the command:

```
DIR | SORT
```

will sort your directory into alphabetical order. The vertical bar causes all output generated on the left side of the bar to be sent to the right side of the bar for processing.

Piping can also be used when you want to send output to a file. If you want your directory sorted and sent to a new file (for example, DIREC.FIL), you could enter:

```
DIR | SORT >DIREC.FIL
```

MS-DOS will create a file named DIREC.FIL on your default drive. DIREC.FIL contains a sorted listing of the directory on the default drive, since no other drive was specified in the command. To specify a drive other than the default drive, enter:

```
DIR | SORT >B:DIREC.FIL
```

This sends the sorted data to a file named DIREC.FIL on drive B.

A pipeline may consist of more than two commands. For example:

```
DIR | SORT | MORE
```

will sort your directory, show it to you one screen at a time, and put **-MORE-** at the bottom of your screen when there is more output to be seen.

Warning

If you use "command piping" do not write protect your default drive diskette.

5. COMMANDS

ABOUT THIS CHAPTER

This chapter gives the syntax and use of all the MS-DOS commands. The commands are presented in alphabetical order.

CONTENTS

INTRODUCTION	5-1	DIR	5-43
MS-DOS 3.20 AND NETWORKING	5-5	DISKCOMP	5-45
COMMANDS	5-8	DISKCOPY	5-48
ASSIGN	5-8	ECHO	5-51
ATTRIB	5-10	ERASE	5-52
BACKUP	5-12	EXE2BIN	5-52
BREAK	5-16	EXIT	5-55
CHDIR	5-17	FC	5-56
CHKDSK	5-19	FDISK	5-66
CLS	5-23	FIND	5-66
COMMAND	5-23	FOR	5-70
COMP	5-27	FORMAT	5-72
COPY	5-30	GOTO	5-76
CTTY	5-36	GRAFTABL	5-78
DATE	5-38	GRAPHICS	5-79
DEL	5-41	GW BASIC	5-82

HEXDUMP	5-83	SHARE	5-131
IF	5-85	SHIFT	5-133
JOIN	5-87	SORT	5-135
LABEL	5-92	SUBST	5-139
MKDIR	5-94	SYS	5-143
MODE	5-95	TIME	5-144
MODE COM:	5-96	TREE	5-146
MODE n	5-98	TYPE	5-147
MODE LPT	5-100	VER	5-149
MORE	5-103	VERIFY	5-149
PATH	5-104	VOL	5-150
PAUSE	5-105	XCOPY	5-151
PRINT	5-107		
PROMPT	5-111		
RECOVER	5-114		
REM	5-115		
REN	5-116		
REPLACE	5-118		
RESTORE	5-122		
RMDIR	5-126		
SELECT	5-127		
SET	5-129		

COMMANDS

INTRODUCTION

The table below lists the commands included in this chapter and gives a brief description of each. It also tells you which commands are internal (I), which are external (E), and which it is not possible to use over a network (U).

COMMAND	CLASS	FUNCTION
ASSIGN	E	Instructs MS-DOS to route all requests for one drive to another drive.
ATTRIBS	E	Sets or resets the read-only attribute of a file. Sets or resets the archive bit attribute of a file.
BACKUP	E	Creates a back up of one or more disk files on a series of disks.
BREAK	I	Turns off and on the abort feature provided by CTRL C and CTRL BREAK
CHDIR	I	Changes the current directory.
CHKDSK	E U	Analyzes the contents of the disk in the specified or default drive.
CLS	I	Clears the screen.
COMMAND	E	Starts a new command processor
COMP	E	Compares the contents of a file or group of files with the contents of another file or group of files.
COPY	I	Copies one or more files to another file or to a device. Alternatively several files can be concatenated and copied to a destination file.
CTTY	I	Changes the input/output console from which you issue commands.

COMMAND	CLASS	FUNCTION
DATE	I	Displays and sets the date known to the system.
DEL	I	Deletes the specified file(s).
DIR	I	Lists the requested directory entries.
DISKCOMP	E U	Compares the contents of two diskettes of the same type.
DISKCOPY	E U	Copies the contents of one diskette onto another diskette.
ECHO	I	Turns the batch file echo feature off and on. Text given as a parameter will be output to the standard output device.
ERASE	I	Is the same as DEL.
EXE2BIN	E	Converts executable files to binary format.
EXIT	I	Exits from a secondary command processor and returns to a parent program or command processor.
FC	E	Compares the contents of two files.
FDISK	E	Sets up MS-DOS partition(s) for the hard disk.
FIND	E	Searches for a specific string of text in file(s).
FOR	I	Allows iterative execution of MS-DOS commands.
FORMAT	E U	Formats a disk to receive MS-DOS files.
GOTO	I	Jumps to a specified position in a batch file.

COMMANDS

COMMAND	CLASS	FUNCTION
GRAFTABL	E	Loads the non-BIOS ASCII characters for graphics modes.
GRAPHICS	E	Enables graphics currently displayed on the screen to be printed on a compatible printer, along with any text when the SHIFT PRT SCR keys are pressed.
GWBasic	E	Enters the MS GW-BASIC interpreter.
HEXDUMP	E	Displays the contents of a file, byte by byte, in hexadecimal.
IF	I	Causes conditional execution of a command in a batch file.
JOIN	E U	Joins a disk drive to an empty directory on another drive to produce a single directory structure.
LABEL	E U	Creates, changes or deletes a disk volume label.
MKDIR	I	Creates a directory.
MODE	E	Sets the monitor mode, serial transmission and printer environments.
MORE	E	A filter which sends output to the terminal one screen at a time.
PATH	I	Sets a command search path.
PAUSE	I	Pauses until a key is pressed in a batch file.
PRINT	E	Queues text files for background printing.
PROMPT	I	Sets the MS-DOS command prompt.

COMMAND	CLASS	FUNCTION
RECOVER	E U	Recovers a file or an entire disk containing faulty blocks.
REM	I	A null command which can be used for putting remarks into a batch file.
REN[AME]	I	Renames files.
REPLACE	E	Replaces previous versions of files.
RESTORE	E	Restores a number of files from back up disks. The back up disks must have been created using the BACKUP command.
RMDIR	I	Removes an empty sub-directory.
SELECT	E	Copies your MS-DOS diskette to create a working copy for your selected country and keyboard.
SET	I	Assigns one string value in the environment to another key string; for use in programs or batch files.
SHARE	E	Installs network file and record locking. Also installs diskette change checking.
SHIFT	I	Allows access to more than the default number of replaceable parameters in batch processing.
SORT	E	A filter, which sorts data alphabetically, in forward or reverse order.
SUBST	E	Substitutes a dummy drive specifier for a pathname.
SYS	E U	Updates the specified disk with the hidden system files, which come from the default drive.

COMMANDS

COMMAND	CLASS	FUNCTION
TIME	I	Displays and sets the system time.
TREE	E	Displays all the directories and paths on the specified drive. It also has an option to list the files in each directory.
TYPE	I	Displays the contents of the specified file on the video screen.
VER	I	This command displays on your screen the version number of the MS-DOS system you are using.
VERIFY	I	Verifies writes to disk.
VOL	I	Displays the volume label of the disk in the specified or default drive.
XCOPY	E	Copies files and subdirectories.

MS-DOS 3.20 AND NETWORKING

MS-DOS 3.20 supports networking using the MS-Network extension software.

The file/record locking mechanism installed when using the following command only works when networking is active.

COMMAND	DESCRIPTION
SHARE	This program loads, then terminates, but stays resident in the Random Access Memory. It installs the file/record locking mechanism.

The following command(s) are useful for networking:

COMMAND	DESCRIPTION
ATTRIB	This command sets or resets the read-only attribute of a file or displays the attributes of that file. If any application opens a file with read/write permission, ATTRIB can set the file to read-only, allowing certain application programs to be run and shared over the network in compatibility mode.
COPY	This command can be used to copy files from a network disk to your own or to another network disk.
DIR	This command can be used to display information about files on network disks.
PRINT	For a printer server, use PRINT with the network printer as a logical device.
REPLACE	This command can be used to update or to add files to network drives. Use this command with the /A switch to restore deleted files, which have been previously backed-up with the XCOPY command.
XCOPY	This command can be used to recursively copy directory structures and the files contained within those directories, from or to network drives. Use this command with the /M switch to carry out incremental backups of network drives; as this switch turns off the archive bit.

COMMANDS

Most MS-DOS commands can be used over the network. But **do not use:**

COMMAND	RESULT...
CHKDSK	the error message: Cannot CHKDSK a Network drive. If you suspect a problem contact the Network Manager.
DISKCOMP	the error message: Cannot DISKCOMP to or from a Network drive. Use COMP *.* for each relevant directory instead.
DISKCOPY	the error message: Cannot DISKCOPY to or from a Network drive. Use COPY *.* for each directory instead.
FDISK	the error message: Cannot FDISK a Network Drive.
FORMAT	the error message: Cannot FORMAT a Network Drive. The Network Manager can stop the server, do FORMAT then restart the server.
JOIN	the error message: Cannot JOIN a Network Drive.
LABEL	the error message: Cannot LABEL a Network Drive.

COMMAND	RESULT...
RECOVER	the error message: Cannot RECOVER to a network drive. The Network Manager can stop the server, do RECOVER, then restart the server.
SUBST	the error message: Cannot SUBST to a network drive.
SYS	the error message: Cannot SYS to a network drive. The Network Manager can stop the server, do SYS then restart the server.
VERIFY	This command will not work for files copied over a network. VERIFY ON will only cause verification of writes to local files.

ASSIGN

Instructs MS-DOS to route all requests for one drive to another drive.

Classification

External

[*d*:][*path*] **ASSIGN** [*drive1* = *drive2*] ...

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where ASSIGN is to be found.
<i>path</i>	Specifies the directory where ASSIGN is to be found.
<i>drive1</i>	The letter of the drive whose requests are to be rerouted.
<i>drive2</i>	The letter of the drive to which requests for drive1 are to be rerouted.

Characteristics

Following execution of the ASSIGN command MS-DOS converts all references for *drive1* to *drive2*.

Note that only the drive letter should be entered in the command line. Do not enter the colon.

If you enter the ASSIGN command without parameters all current assignments will be reset.

Examples

IF you enter...	THEN...
ASSIGN A = C B = C	all requests to drives A or B will be rerouted to drive C.
ASSIGN	all assignments are reset.

Remarks

Never ASSIGN drives and then use the following commands or unpredictable results will occur and/or error messages will be displayed.

BACKUP DISKCOMP DISKCOPY FORMAT JOIN LABEL PRINT
RESTORE SUBST

ATTRIB

Sets or resets the read-only attribute and/or archive bit attribute of a file.

Classification

External

`[d:][path] ATTRIB [+R |-R] [+A |-A] [drive:][file-path]filename`

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where ATTRIB is to be found.
<i>path</i>	Specifies the directory where ATTRIB is to be found.
<i>drive</i>	Specifies the drive where filename is to be found
<i>file-path</i>	Specifies the directory where filename is to be found.
<i>filename</i>	The filename of the files you want to reference. Wildcard characters (* and ?) can be used in the filename.

Characteristics

- + R sets the read-only attribute of a file.
- R disables read-only mode.
- + A sets the archive attribute of a file.
- A clears the archive attribute of a file.

To display the attribute of files enter:

ATTRIB *filename*.

Remarks

If an application opens a file with read and write permission, ATTRIB forces read-only mode to allow file sharing over a network.

The BACKUP, RESTORE, and XCOPY commands use the archive attribute to control a selective Backup/Restore/Xcopy on files that have been modified. You can use the +A and -A options to select files that you want to back up with the BACKUP /M switch or copy with the XCOPY /M switch.

Example

The following example makes the file named MYFILE.TXT read-only:

```
ATTRIB +R MYFILE.TXT
```



BACKUP

Creates a backup of one or more disk files on a series of disks. The source is usually a hard disk, but can be a floppy disk. The target is usually a floppy disk, but can be a hard disk. However the source and the target must be different drives.

Classification

External

[d:][path] BACKUP source-drive:[pathname] target-drive: [/S] [/M] [/A] [/D: date]

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where BACKUP is to be found.
<i>path</i>	Specifies the directory where BACKUP is to be found.
<i>source-drive</i>	The disk drive to be backed up.
<i>pathname</i>	The files you wish to back up. If you enter only the source-drive specifier then only those files in the current directory are backed up. If you specify a path terminating in a directory name then all files in that directory will be backed up. If the path terminates in a file name (or a group of file names specified using wild card characters) only the specified file(s) will be backed up.
<i>target-drive</i>	The disk drive in which the backup copy is to be made.
<i>/S</i>	The files contained in the subdirectories are to be backed up as well as those in the specified directory (or current directory if no directory is specified). This includes files at all directory levels below the specified directory.
<i>/M</i>	Only those files in the specified directory that have been modified or created since the last backup will be backed up.

SYNTAX ELEMENT	MEANING
/A	The specified files will be added to the disk already inserted in the diskette drive. If /A is not specified you will be prompted to insert a diskette once the BACKUP program is in memory.
/D:date	Only those files in the specified (or current) directory that have been created or modified since the specified date are to be backed up. Refer to the DATE command for valid date formats.

Characteristics

Once you have entered the BACKUP command a prompt will be issued asking you to insert a target diskette (unless you included /A in the command line). You must use MS-DOS formatted diskettes. Any files that already existed on the target diskette will be deleted unless you used the /A option. Once the target diskette is full you will be prompted to insert another target diskette. Be sure to label each diskette as the order will be important when you restore your backup to hard disk.

As each file is backed up its name is displayed on the screen.

The exit code is set by the BACKUP command as follows:

- 0 Normal completion.
- 1 No files found.
- 2 Some files not backed up due to file sharing conflicts.
- 3 Command execution terminated by the user.
- 4 Command execution terminated due to an error.

The error level exit code can be used by the batch processing IF ERRORLEVEL command.

COMMANDS

Note

The files on the backup diskettes cannot be used except for restoring using the RESTORE command.

Warning

You should not use the BACKUP command if the drive you are backing up has been ASSIGNED, JOINed, or SUBSTITuted. If you do, you may not be able to restore the files with the RESTORE command.

Examples

IF you enter...	THEN...
BACKUP C:*.COM A:	each file with the .COM extension in the current directory of the hard disk drive C is backed up onto a series of diskettes in drive A.
BACKUP C:*. * A:/S	all files on the hard disk drive C will be backed up onto a series of diskettes in drive A.
BACKUP C:MYDIR\ MYFILE A: /A	the file named MYFILE in the directory MYDIR is added to the backup diskette in drive A.
BACKUP *. * A: /M	all files in the current directory on the hard disk that have been created or modified since the last backup was made are backed up onto a series of diskettes in drive A.
BACKUP *. * A: /D:01-01-84	all files in the current directory on drive C that have been created or modified since 1 January 1984 are backed up onto a series of diskettes in drive A.



BREAK

Turns off and on the abort feature provided by **CTRL C** and **CTRL BREAK**.

Classification

Internal

BREAK [ON | OFF]

Characteristics

The default setting of **BREAK** is **OFF**. With this setting MS-DOS normally checks **CTRL BREAK** only for input/output operations involving screen, keyboard and printer. **CTRL BREAK** will not normally abort other functions, such as the assemble or compile operations. Specifying **BREAK ON** makes **CTRL BREAK** effective for other functions.

If you specify **BREAK** without a parameter the current **BREAK** setting is displayed.

Changes the current directory; displays the name of the current directory.

Classification

Internal

Syntax 1

CHDIR [*drive:*][*path*]

Syntax 2

CD [*drive:*][*path*]

Where

SYNTAX ELEMENT	MEANING
<i>drive</i>	The letter of the drive where the new directory is to be found.
<i>path</i>	A path that terminates with the name of the directory you wish to enter.

Characteristics

Use CHDIR (or CD) with a path to change to any valid directory.

To change to the parent directory of your current directory enter:

CHDIR ..

Used without a parameter CHDIR displays the full path and name of your current directory. This feature is useful if you have forgotten the name of the directory in which you are working.

To change to the root directory enter:

CHDIR \

Examples

IF you enter...	THEN...
CHDIR \BIN\USER\JOE\FORMS	MS-DOS puts you in the directory BIN\USER\JOE\FORMS
CHDIR ..	MS-DOS puts you in the parent of the current working directory (in the above example, \BIN\USER\JOE)

Analyzes the contents of the disk in the specified or default drive.

Classification

External, Non-network

[d:][path] **CHKDSK** *[drive:] [dir-path][filename] [/F] [/V]*

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where CHKDSK is to be found.
<i>path</i>	Specifies the directory where CHKDSK is to be found.
<i>drive</i>	The drive containing the disk to be checked.
<i>dir-path</i>	The path to the directory to be checked.
<i>filename</i>	A file or group of files specified using wild card characters. A status report for the individual files will be displayed, if they are non-contiguous.

SYNTAX ELEMENT	MEANING
/F	CHKDSK tries to correct any errors it finds.
/V	CHKDSK displays status messages for each directory, subdirectory and each file specified.

Characteristics

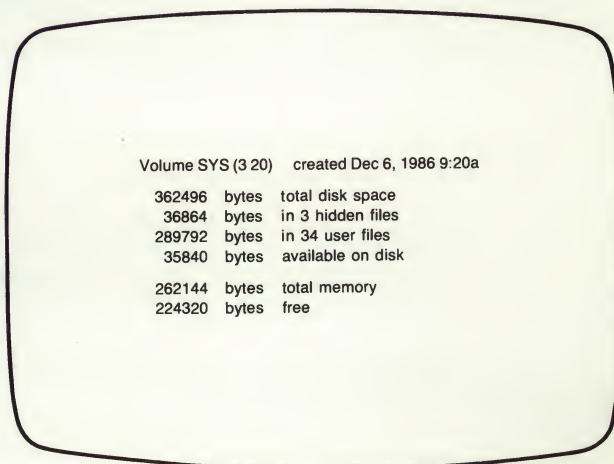
CHKDSK produces a status report on the File Access Tables, Directories, Files, and bad sectors of the disk. Run CHKDSK with the /F switch to attempt to correct these error then the status report informs you of faults with files and/or directories, the cause may be bad sectors on the disk. If so the first time you run CHKDSK with the /F switch the problems with the files and directories should be fixed, this may release some space on the disk. Run CHKDSK with the /F switch a second time to mark faulty clusters in this freed space. A cluster on the disk consists of one or more sectors, depending on the cluster and sector size.

If you specify the /V switch, CHKDSK displays messages while it is running and will also list the hidden files.

If no parameter is specified then the disk in the default drive is checked.

Example

The following is an example of a CHKDSK status report:



```
Volume SYS (3 20)   created Dec 6, 1986 9:20a

362496 bytes total disk space
36864 bytes in 3 hidden files
289792 bytes in 34 user files
35840 bytes available on disk

262144 bytes total memory
224320 bytes free
```

Fig. 5-1 CHKDSK Example

Remarks

To redirect CHKDSK output, use the > redirection symbol followed by the name of a file. Error messages will be sent to the file specified. Do not use the /F switch if you redirect CHKDSK output.

If you use the /F switch CHKDSK tries to fix any errors found in the directory, file or file allocation table (FAT). When errors are found, due to lost clusters, CHKDSK asks you "**Convert lost chains to files (Y/N)?**". If you reply **Y** then press **ENTER**, CHKDSK recovers each cluster in the disconnected chain to a file called **FILEnnnn.CHK**, in the root directory of the specified drive (where **nnnn** starts at 0000 and increases by 1 for each lost chain). See Appendix D "ERROR MESSAGES" for a list of all the messages CHKDSK issues.

Space on diskettes is allocated in clusters. Diskettes that have had a lot of file creation and deletion activity become fragmented, because clusters are not allocated sequentially. The first free cluster found is the next cluster allocated regardless of its location on the diskette.

A fragmented diskette can cause poor performance due to delays involved in reading or writing a file.

CHKDSK will display one of the following messages:

filename contains non-contiguous blocks

or

All specified file(s) are contiguous

If the first message appears and you are experiencing poor disk performance, use the COPY command to copy all the files in the directory to a newly formatted diskette. Then use the copy rather than the original.

Use the command CHKDSK *.* to find out whether the files in the current directory are contiguous or not.

CLS

Clears the screen.

Classification

Internal

CLS

Characteristics

All data on the display screen is cleared. The cursor is moved to the upper left hand corner (the home position). This command has no effect on memory or files.

COMMAND

Starts a new command processor

Classification

External

`[d:][path] COMMAND [drive:shellpath][cttydev] [/E: nnnnn]
[/P] [/C command-string]`

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where COMMAND is to be found.
<i>path</i>	Specifies the directory where COMMAND is to be found.
<i>drive</i>	A single letter drive-name of the drive containing COMMAND.COM.
<i>shellpath</i>	The pathname of the directory containing COMMAND.COM
<i>cttydev</i>	An alternative device for standard input and output (see the CTTY command for more details).
<i>/E:nnnnn</i>	This switch specifies the environment size, where <i>nnnnn</i> is the size in bytes. The size may range between 160 and 32768 bytes. The default value is 160 bytes.
<i>/P</i>	This switch makes this copy of COMMAND.COM permanent. It is not possible to exit to the primary command processor without re-booting the system. If the AUTOEXEC.BAT file is present in the root directory of the current drive, it is executed; otherwise you are prompted for the current date and time.

COMMANDS

SYNTAX ELEMENT	MEANING
<i>/C command-string</i>	This switch causes the <i>command-string</i> to be passed to the new command processor for execution and then exit to the original command processor. This must be the last switch if used; everything following is part of the <i>command-string</i> .

Characteristics:

COMMAND.COM is loaded into memory in two parts: the transient part and the resident part. Some application programs write over the transient part of the command processor when they run. When this happens the resident part looks into the COMSPEC variable in the environment, to find the command processor file.

COMSPEC = *drive:shellpath*

typically

COMSPEC = A:\COMMAND.COM

You can use SET without any parameters to examine the environment. (See SHELL in Appendix C for details on loading the top level command processor). One application of this is to install a copy of COMMAND.COM on Virtual Disk using (VDISK); calling it D:, for example. Then to issue the following call to invoke the second command processor.

D:\COMMAND D:\ /P

Another application is to call COMMAND without any parameters or only with *drive:shellpath*. This invokes a secondary command processor (a child), which inherits the parent command processor's environment and prompt. If this environment and/or prompt is modified, these changes are only known to the child processor and its applications. Using EXIT (see EXIT command in this chapter), reinstates the parents environment. For example at the A> prompt enter:

COMMAND

The computer will respond:

```
Microsoft(R) MS-DOS(R) Version 3.20
(C)Copyright Microsoft Corp 1981-1986
A>
```

enter:

```
PROMPT time = $T$G
```

the computer will respond with the prompt incorporating the time known to the computer:

```
time = 17:10:11.80>
```

If you now enter:

```
EXIT
```

the computer exits to the parent processor which has the prompt.

```
A>
```

In batch files calling

```
COMMAND /C batch__filename [parameter....]
```

enables the original parent batch file to call a child batch file and command processor as a subroutine. Upon completion of the child batch file, there is an automatic exit to the parent batchfile and command processor.

Normally MS-DOS allocates 160 bytes (10 main memory paragraphs) for the environment table. This may not be enough if you want to set numerous environment variables using the SET or PATH command. A subdirectory can have a pathname of up to 63 characters, each character uses one byte. So with environment variables set to point to long pathnames, you could easily require more than 160 bytes. With MS-DOS Ver. 3.20 it is now possible to increase the environment size of the command processor.

If you specify the environment size to be less than 160 bytes, the new command processor will have an environment of 160 bytes and the following error message is issued:

Invalid environment size specified

If you specify the environment size to be greater than 32768 bytes, the new command processor will have an environment of 32768 bytes and the above error message is issued. Between these two limits environment space is allocated in paragraphs, each paragraph consists of 16 bytes. When you specify an environment size, which is not a multiple of 16 the actual environment size is rounded up to be a multiple of 16, that is to the next paragraph boundary.

COMP

Compares the contents of a file or group of files with the contents of another file or group of files. This is useful for checking the results of a COPY operation.

Classification

External

`[d:][path] COMP [pathname1 [pathname2]]`

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where COMP is to be found.
<i>path</i>	Specifies the directory where COMP is to be found.
<i>pathname1</i>	The file or group of files (specified using wild card characters) that the file or files specified by <i>pathname2</i> are to be compared with. If the path terminates in a directory name, all files in the specified directory are compared. If the path terminates in a drive specifier the files in the current directory of that drive are compared.
<i>pathname2</i>	The file or group of files (specified using wild card characters) to be compared with the file or group of files specified by <i>pathname1</i> . If the path terminates in a directory name or drive specifier, only the files with the same file name as those specified by <i>pathname1</i> will be compared.

Characteristics

The files you wish to compare may be on the same or different drives, or in the same or different directories.

If you enter the command without parameters, or if you omit the second parameter, you will be prompted for the missing parameters.

As the COMP command proceeds it displays the files and paths of the files currently being compared. An error message is issued if a specified directory path is invalid, or the two files to be compared are different in size, or a file specified by *pathname2* cannot be found.

COMMANDS

If the same location in two files contains information which does not match, a message is issued indicating the offset (in bytes) within the files and the contents of the bytes for each file. If ten such mismatches are found the comparison terminates and issues an appropriate message.

If, at the end of a comparison, the *end-of-file* marker cannot be found, the following message will be displayed:

EOF mark not found

This is necessary as some applications create files that are always recorded in multiples of 128 bytes, although the data actually occupies a few bytes less than stated in the directory. The COMP command may therefore find compare errors in the last 128 bytes beyond the last actual data byte. The above message therefore indicates that the actual data in the two files matches.

Following a successful comparison COMP displays a message indicating that the files match and continues comparing the next pair of files. After all the specified files have been compared the following prompt appears:

Compare more files (Y/N)?

Press N to terminate the command, or Y if you wish to compare more files. If you press Y you will be prompted for the files you wish to compare.

Examples

IF you enter...	THEN...
COMP A:*.LST B:*.CPY	all the files on drive A with the extension LST are compared with the files of the same name but with extension CPY on drive B
COMP A:*.LST C:	all the files on drive A with the extension LST are compared with the files of the same name on the current directory of drive C

COPY

Copies one or more files. Alternatively several files can be concatenated and copied to a target file.

Classification

Internal

COPY [/A|/B] *pathname1* [/A|/B] [+ *pathname2* [/A|/B]...] [*pathname* [/A|/B]][/V]

Where

SYNTAX ELEMENT	MEANING
<i>pathname1</i>	The path of the file to be copied (excluding the drive only if the file is on the default drive, excluding the directory path only if the file is in the current directory).
<i>pathname2</i>	The path of any file to be concatenated with the file in <i>pathname1</i> (excluding the drive specifier only if the file is in the default drive, excluding the directory path only if the file is in the current directory).
<i>pathname</i>	The path of the target directory or file (excluding the drive to place the file in the default drive, or the file name to retain that given in <i>pathname1</i>).
/V	Verify the target file by a read after a write and COPY then compares this data. An error message is output if this comparison fails

COMMANDS

Characteristics

COPY either makes a copy of a file or concatenates two or more files. The resulting target file may have the same name as the first source file (so long as it is in another directory), or a different name.

To copy a file and retain its name, use the following syntax:

COPY *pathname1* [*pathname*]

where *pathname* gives the drive and/or directory in which the file is to be placed.

Omit the *pathname* parameter to place the copy in the current directory of the default drive. Specify simply a drive to place the copy in the current directory of that drive. If you specify a directory path, make sure that you end the path with the name of an existing directory.

Two files cannot have the same name in the same directory: if you try to copy a file onto itself you will get an error message.

To copy a file and give the copy a different name, use the following syntax:

COPY *pathname1* *pathname*

Where *pathname* gives the name of the file in which the copy is to be placed.

Omit drive and directory names from *pathname* to place the file in the current directory of the default drive. Omit just directory names from *pathname* to place the file in the current directory of the specified drive. Because you are renaming the file, the source and target directories may be the same.

If you specify in *pathname* a file that does not exist, MS-DOS creates it for you. If the file already exists, its previous contents are destroyed in the copy operation (**warning**: before copying be sure that you do not want these contents).

To concatenate two or more files, use the following syntax:

COPY *pathname1* + *pathname2* [+ *pathname3*] ... [*pathname*]

Combine the names of each file to be concatenated with the plus sign (+). COPY appends each file in turn to the previous one.

The result of the concatenation is a single file. This file is given the drive, directory, name and extension specified in *pathname*; if this is absent, the resulting file replaces the file specified in *pathname1* (that is, the first file to be concatenated). If only the drive is supplied, a new file with the name of the file specified in *pathname1* is copied into the drive's current directory. If drive and directory names only are supplied, a new file with the name given in *pathname1* is copied into the specified subdirectory. If only the name and extension are supplied, the file is placed in the current directory of the default drive.

Examples

IF you enter...	THEN...
COPY B:SECRETS	the file SECRETS is copied from drive B to the default drive.
COPY *.* B:	all files in the current directory on the default drive are copied to drive B.
COPY \DOCS\SECRETS B:INFO	the file SECRETS is copied to the file INFO in the current directory of drive B (or the subdirectory INFO in the current directory if it exists).

COMMANDS

IF you enter...	THEN...
COPY \DOCS\SECRETS \DOCS\NEWS	the file SECRETS is copied to the file NEWS (or the subdirectory NEWS if it exists).
COPY INFO + NEWS + VIEWS ALL.LST	the files NEWS and VIEWS are appended to the file INFO, and the resulting concatenation is copied to the file ALL.LST.
COPY ALL.LST + NEWS	the file NEWS is appended to the file ALL.LST, resulting in an enlarged ALL.LST.
COPY *.LST COMBIN.PRN	all files with the extension .LST are concatenated, and the result is placed in COMBIN.PRN. The source and target are operated on as ASCII files, by default.
COPY *.LST + *.REF *.PRN	each file with the extension .REF is appended to the file with the same name but the extension .LST, and each resulting file is given the extension .PRN
COPY *.LST + *.REF COMBIN.PRN	all files with the extension .LST then all files with the extension .REF are placed in COMBIN.PRN.
COPY ALL.LST + *.LST	all files with the extension .LST, with the exception of ALL.LST, are appended to ALL.LST.
COPY PROG.COM/B + ERRS.TXT/A	the text file ERRS.TXT is appended to the binary file PROG.COM, leaving the result in the binary file PROG.COM.

Remarks

As the examples show, you can use the wild cards * and ?, to do both simple copy operations (where no target file is specified) and file concatenation.

When wild cards are present in two or more source parameters combined with the concatenation symbol (+), the result is a single target file. However if the target parameter itself contains a wild card, a series of concatenated files are produced.

Note that the penultimate example shows the correct way of concatenating files where one of the source files is also the target file. Had the command `COPY *.LST ALL.LST` been entered, the previous contents of `ALL.LST` would have been destroyed and the following message would have appeared:

Content of destination lost before copy

You may use the following reserved device names in place of standard file name parameters (the colons are optional):

AUX:	LPT2:
COM1:	LPT3:
COM2:	NUL:
CON:	PRN:
LPT1:	

For example, to copy text that you are about to enter from the keyboard into a file, use the following format:

`COPY CON: filename`

You can then enter text directly into the file you have named, terminating your input by pressing **CTRL Z** followed by **ENTER**.

The parameters `/A` and `/B` shown in the syntax of the `COPY` command apply when you wish to regulate the amount of data to be copied. The following table shows the effect of `/A` and `/B` on files to which they are attached and on all remaining files in the command until another parameter is found.

COMMANDS

IF you enter...	WITH...	THEN...
/A	a source file	the file is regarded as a text (ASCII) file, and its contents copied up to but excluding the first end-of-file character (CTRL-Z). This is the default for concatenation.
/A	a target file	the file is regarded as a text (ASCII) file, and an end-of-file character (CTRL-Z) is added as its last character. This is the default for concatenation.
/B	a source file	the file is regarded as a binary file, and the entire file including any number of end-of-file characters is copied. This is the default for simple copy.
/B	a target file	the file is regarded as binary, and no end-of-file character (CTRL-Z) is added. This is the default for simple copy.

The default value is /A when you are using COPY to concatenate files, /B when you are using COPY simply to copy files.

See the last example COPY PROG.COM/B + ERRS.TXT/A which shows the use of /A and /B to append a file of error messages to a program file. The default for concatenated files being /A the /B attached to the program file is obligatory. The /A must then be attached to the text file in order to cancel the previous /B parameter.



CTTY

Changes the input/output console from which you issue commands.

Classification

Internal

CTTY *device*

Where

SYNTAX ELEMENT	MEANING
<i>device</i>	The reserved name of the device you wish to use.

Characteristics

A suitable terminal must be connected to the device port. Command Input/ Output is passed to the alternate terminal. The CTTY CON command must be entered at the alternate terminal to restore input/output back to the normal console.

COMMANDS

Examples

IF you enter...	THEN...
CTTY AUX:	command I/O is moved to the device attached to the RS-232-C.
CTTY CON:	command I/O is returned to the console.

Note

You must use MODE to initialize the device before use. There are many programs that do not use MS-DOS for input and/or output, but use the BIOS or hardware ports. The CTTY command will have no effect on these programs. CTTY will only affect programs that use MS-DOS.



DATE

Displays and sets the date known to the system.

Classification

Internal

Syntax 1 USA

DATE [*mm-dd-yy*]

or

Syntax 2 Europe

DATE [*dd-mm-yy*]

Where

SYNTAX ELEMENT	MEANING
<i>mm</i>	The one or two-digit identifier of the month (1-12).
<i>dd</i>	The one or two-digit identifier of the day (1-31).
<i>yy</i>	The two or four-digit identifier of the year (80-99 or 1980-2099).

Characteristics

The syntax depends on the COUNTRY setting in CONFIG.SYS.

Separate month, day and year entries by either hyphens (-) or slashes (/).

If you leave out the parameter, DATE prompts you as in the following example:

Current date is Tue 11-5-85
Enter new date:

Enter the date in the correct format, without entering a value for the day of the week. To accept the current date simply press **ENTER**.

Example

IF you enter...	THEN...
DATE 1-2-84	2nd January 1984 is established as the current date.
DATE	the DATE program prompts you to enter the date.

Remarks

If the values or separators you enter are not valid, DATE displays the message:

Invalid date

Enter new date:

DATE then waits for you to enter a valid date.

COMMANDS

DEL

Deletes the specified file(s).

Classification

Internal

Syntax 1

DEL [*drive:*] *pathname*

Syntax 2

ERASE [*drive:*] *pathname*

Where

SYNTAX ELEMENT	MEANING
<i>drive</i>	Specifies the drive where the file(s) to be deleted are to be found.
<i>pathname</i>	The specification of the file(s) to be deleted, excluding the directory path to delete file(s) in the current directory.

Characteristics

You may use the wild cards * and ? in the file name and extension.

To delete all the files in a directory enter the wild cards "*.*". Alternatively, enter a path ending in a directory. In these cases, MS-DOS prompts you to confirm your choice:

Are you sure (Y/N)?

Press **Y** to carry out the deletion, or **N** to return to the MS-DOS prompt.

To delete all files without a file extension, enter "*.*".

Example

IF you enter...	THEN...
DEL B:*.TMP	all files in the default directory on drive B: with the extension .TMP are deleted.

Remarks

To delete an actual directory (as opposed to all the files that a directory contains) you must use the RMDIR command, but first you must delete all the files that directory contains.

The command ERASE is synonymous with DEL.

Lists details of the files in the current or a specified directory.

Classification

Internal

DIR [*drive:*][*pathname*] [/P] [/W]

Where

SYNTAX ELEMENT	MEANING
<i>drive</i>	Specifies the drive where the file and directory names to be listed are to be found.
<i>pathname</i>	The specification of the file and directory names to be listed, excluding the directory path to list names in the current directory.
/P	The directory display halts as soon as the screen becomes full. Press any key to resume the listing.
/W	File and directory names only are displayed, five to a line across the screen.

Characteristics

If you do not specify a file name and extension, all files in the specified (or current) directory are listed.

You may use the wild cards * and ? in the file name and extension. If you omit either the name or the period and extension, the wild card * is assumed in its place.

To list a file that does not have an extension but exclude any that do in the pathname, enter the file name followed by a period (.).

DIR produces a display in which the size in bytes and date and time of creation or last modification appear alongside the file name(s).

Remarks

DIR does not display hidden files in a directory.

Examples

IF you enter...	THEN...
DIR B: (or DIR B:*.*)	all files in the current directory on drive B are listed.
DIR .COM (or DIR *.COM)	all files in the current directory on the default drive having the extension COM are listed.
DIR AUTHORS (or DIR AUTHORS.*)	all files in the current directory on the default drive with the name AUTHORS and any extension are listed.

Compares the contents of two diskettes of the same type.

Classification

External

[*d*:][*path*] **DISKCOMP** [*drive1*: [*drive2*:]] [/1] [/8]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where DISKCOMP is to be found
<i>path</i>	Specifies the directory where DISKCOMP is to be found.
<i>drive1</i>	The drive to contain the first of the two diskettes to be compared.
<i>drive2</i>	The drive containing the diskette to be compared with the diskette in <i>drive1</i> .
/1	Only the first sides of the diskettes are compared.
/8	Only eight sectors per track are compared, even if the diskette in <i>drive1</i> has nine sectors per track.

Characteristics

The DISKCOMP command can only be used to compare diskettes. If you specify a hard disk drive an error message will be issued.

If neither drive is specified a single drive comparison is done on the default drive.

If only *drive1* is specified then *drive2* assumes the default drive.

The DISKCOMP command prompts you to insert the diskettes at the appropriate time then waits for you to strike any key before continuing.

The DISKCOMP command compares all tracks on the diskettes and indicates mismatched tracks by track and side number.

When using DISKCOMP to compare diskettes in different drives, those drives must be of the same type.

The DISKCOMP command determines the number of sides to be compared and the number of sectors per track from the first diskette. That is:

- If the first diskette is High Density 1.2 MB (80 tracks, 15 sectors and double sided) then the second diskette must also be High Density 1.2MB.
- If the first diskette is 720KB (80 tracks, 9 sectors and double sided) then the second diskette must also be 720KB with the same specification.
- If the first diskette is dual-sided and has nine sectors per track then a nine-sectors-per-track comparison on both sides of the second diskette will be performed (unless /1 and/or /8 was specified). If the second diskette is single-sided or formatted eight sectors per track an error message will be displayed.
- If the first diskette is single-sided then only the first side of the second diskette will be compared regardless as to whether the second diskette is single or double sided.

COMMANDS

- If the first diskette is formatted eight sectors per track, then only eight sectors per track of the second diskette will be compared regardless as to whether the second diskette is formatted eight or nine sectors per track.

When all tracks have been compared the following prompt appears:

Compare more diskettes (Y/N)?

Press **Y** to perform another comparison using the same drives, or press **N** to exit the program.

Examples

IF you enter...	THEN...
DISKCOMP A: B:	the diskette in drive B is compared with the diskette in drive A.
DISKCOMP /1	a single-drive comparison is done using the default drive. Only the first sides of the diskettes are compared.
DISKCOMP B: /8	an eight-sectors-per-track comparison is done between the diskettes in drive B and the default drive. If drive B is the default drive then a single-drive comparison is done.

Remarks

The DISKCOMP command compares entire diskettes. If you wish to compare only files you must use the COMP command.

The DISKCOMP command cannot be used to compare a diskette created using the COPY command with the original because the COPY command copies on a file-by-file (not track-by-track) basis.

On single-drive systems all prompts are for drive A, regardless of the drive letters you entered in the command line.

DISKCOPY

Copies the contents of a diskette in one drive onto another diskette.

Classification

External, Non-network

[*d:*][*path*] **DISKCOPY** [*sourcedrive:*] [*targetdrive:*] [/1]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where DISKCOPY is to be found
<i>path</i>	Specifies the directory where DISKCOPY is to be found.
<i>sourcedrive</i>	The letter of the drive that contains the diskette to be copied.
<i>targetdrive</i>	The letter of the drive that contains the diskette to receive the copy.
/1	This switch specifies that only the first side of the diskette is to be copied.

COMMANDS

Characteristics

The DISKCOPY command copies entire diskettes. Use COPY to copy files, or to copy to a different disk type than the source. For DISKCOPY, the diskettes must be of the same type, i.e. single, double, or quadruple density or high capacity. Also the source drive and the target drive must be able to read and write diskettes of the same type (see Chapter 1 for a table of Diskette and Drive compatibility). DISKCOPY automatically determines the number of sides to copy, based on the source drive and diskette. The target diskette is formatted or reformatted if necessary, during the copying. You can use the CHKDSK command to determine the capacity and DISKCOMP A: A: (comparing a disk with itself) to determine the format of the source diskette.

If you have 3 1/2 inch disk drives as drive A: and drive B: the operating system needs configuring to handle this media. Place the following declarations in your CONFIG.SYS:

```
DRIVPARM = /D:0 /F:2
```

```
DRIVPARM = /D:1 /F:2
```

(/D:0 indicates drive A:, /D:1 indicates drive B:, /F:2 indicates 720KB capacity.) After changing the CONFIG.SYS file your computer must be re-bootstrapped. See the MS-DOS Software Installation Guide for more details.

With DISKCOPY you can specify the same drives or you may specify different drives. If the drives designated are the same, a single-drive copy operation is performed. You are prompted to insert the disks at the appropriate times. DISKCOPY waits for you to press any key before continuing. If you omit both parameters, a single-drive copy operation will be performed on the default drive. If you omit the second parameter the default drive will be used as the target drive.

After copying, DISKCOPY prompts:

```
Copy complete  
Copy another disk (Y/N)?
```

If you press Y, the next copy is performed on the same drives that you originally specified. You are prompted to insert the proper diskettes.

To end the copy, press N.

Error Codes

The following error codes are returned by DISKCOPY, these can be tested by IF ERRORLEVEL in a batch file.

- 0 Copied Successfully. The last diskcopy was completed with no errors.
- 1 Non-fatal read/write error. An un-recoverable but non-fatal read or write error occurred.
- 2 CTRL C error. The user entered CTRL C to terminate DISKCOPY.
- 3 Fatal hard error. DISKCOPY was unable to read the source disk or format the target disk.
- 4 Initialization error. There is not enough memory or the DISKCOPY command line syntax is incorrect or an invalid drive was specified.

Remarks

After an apparently successful DISKCOPY, you can carry out a DISKCOMP to compare the source and target diskettes.

If diskette errors are encountered during a DISKCOPY, you can run CHKDSK with the /F switch to try to correct errors on the source diskette. Use XCOPY or COPY *.* for each directory, instead of DISKCOPY to copy the suspect diskette.

Do not use DISKCOPY when a directory on the source disk is JOINED to another drive. DISKCOPY does not acknowledge an ASSIGNED drive: the DISKCOPY parameters refer to physical drives.

Turns the batch file echo feature off or on or outputs a message to the standard output device.

Classification

Internal

ECHO [**ON** | **OFF**] *message*

Characteristics

Normally, commands in a batch file are displayed ("echoed") on the screen when they are interpreted by the command processor. ECHO OFF turns off this feature. ECHO ON turns the echo back on.

If ON or OFF is not specified, then the current setting is displayed.

ECHO *message* outputs the message. Note that this message can be redirected using (> or >>). However when appending to a file, problems can arise if the original file is delimited by an End-Of-File (CTRL Z).

Remarks

ECHO *message* can be used as a command outside of batch files.



ERASE

Deletes the specified file(s)

See the "DEL" command.



EXE2BIN

Converts files from .EXE format to binary format.

Classification

External

`[d:][path] EXE2BIN pathname1 [pathname2]`

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where EXE2BIN is to be found.
<i>path</i>	Specifies the directory where EXE2BIN is to be found.
<i>pathname1</i>	The file specification of the file to be converted (excluding the drive if it is in the default drive, excluding a directory path if it is in the current directory, excluding the extension to use the default extension of .EXE).
<i>pathname2</i>	The file specification of the output file (excluding the drive and/or file name to accept the drive and/or file name in <i>pathname1</i> excluding a directory path to accept the current directory, excluding the extension to use the default extension of .BIN).

Characteristics

The input file must be in valid .EXE format produced by the linker. The resident, or actual code and data part of the file must be less than 64K. There must be no STACK segment.

Two kinds of conversions are possible, depending on whether the initial CS:IP (Code Segment: Instruction Pointer) is specified in the .EXE file:

1. If CS:IP is not specified in the .EXE file, a pure binary conversion is assumed. If segment fixups are necessary (that is, the program contains instructions requiring segment relocation), you will be prompted for the fixup value. This value is the absolute segment at which the program is to be loaded. The resultant program will be

usable only when loaded at the absolute memory address specified by a user application. The command processor will not be able to load the program.

- 2 If CS:IP is 0000:100H, it is assumed that the file will run as a .COM file with the location pointer set at 100H by the assembler statement ORG; the first 100H bytes of the file are deleted. No segment fixups are allowed, as .COM files must be segment relocatable; that is, they must assume the entry conditions explained in the MS-MACRO ASSEMBLER User Guide. Once the conversion is complete, you may rename the output file with a .COM extension. Then the command processor will be able to load and execute the program in the same way as the .COM programs supplied on your MS-DOS disk.

Remarks

If the input file does not meet one of the two requirements given above, the following message appears:

File cannot be converted

Note that to create a standard .COM file using the assembler you must set the location pointer at 100H using the ORG statement and use the END statement to set the first location as the start address. For example:

```
ORG      100H
START:
.
.
.
END      START
```

Do not have a .EXE file and a .COM file of the same name in the same directory, when you execute the file.

Exits from a secondary command processor and returns to a parent program or command processor.

Classification

Internal

EXIT

Characteristics

This command can be used when you are running a program and have started a secondary MS-DOS command processor, then want to return to your program. For example, to look at a directory on drive B: while running GW-BASIC, you must start the command processor by entering SHELL. The system prompt will appear.

A>

You can now enter the DIR command and MS-DOS will display the directory. When you enter EXIT, you return to the parent, GW-BASIC.

Note

The error level exit code if set by a program, is returned to the calling program. For example, the following batch script calls a secondary command processor:

```
A> Type COM.BAT
COMMAND
IF ERRORLEVEL 1 ECHO Error
```

When you invoke this batch file by entering COM, the following message will appear:

**Microsoft(R) MS-DOS(R) Version 3.20
(C)Copyright Microsoft Corp 1981-1986**

A>

enter for example:

FORMAT B:

When formatting is complete enter: EXIT

If format returns an errorcode of 1 or greater, then the original batch file will echo:

Error



FC

Compares the contents of two files.

Classification

External

**[d:][path] FC [/a] [/b] [/c] [/l] [/lb length] [/n] [/t] [/w] [/#] filename1
filename2**

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where FC is to be found.
<i>path</i>	Specifies the directory where FC is to be found.
<i>filename1</i>	The name of the first file to be compared.
<i>filename2</i>	The name of the second file to be compared.

Characteristics

The File Comparison utility, FC, compares the contents of two files. The differences between the two files can be output to the screen or to a third file. The files being compared may be either source files (files containing source statements of a programming language) or binary files (output from the assembler, the MS-LINK Linker utility, or a high-level language compiler).

The comparisons are either on a line-by-line or a byte-by-byte basis. The line-by-line comparison isolates blocks of lines that are different between the two files and prints those blocks of lines. The byte-by-byte comparison displays the bytes that are different between the two files.

The following table describes the switches available with the File Comparison utility. It is important to enter the switches in lower case: upper case switches are not recognized.

SWITCH	MEANING
/a	<p>Abbreviates the output of an ASCII comparison. Instead of displaying all the lines that are different, only the lines that begin each set of differences are displayed. The intermediate lines are represented by ellipsis (...)</p>
/b	<p>A binary comparison of both files is performed. The two files are compared byte-to-byte, with no attempt to re-synchronize after a mismatch. The mismatches are printed as follows:</p> <pre>--ADDRS----F1----F2 xxxxxxx yy zz</pre> <p>(where xxxxxxxx is the relative address of the pair of bytes from the beginning of the file). Addresses start at 00000000; yy and zz are the mismatched bytes from file1 and file2, respectively. If one of the files contains less data than the other, then a message is printed out. For example, if file F1 ends before file F2, then the utility displays:</p> <p>fc: F2 longer than F1. This option is the default when you compare files with extensions of ".EXE", ".COM", ".SYS", ".OBJ", ".LIB" or ".BIN".</p>
/c	<p>The utility ignores the case of letters. All letters in the files are considered upper case letters. For example, (note that an underscore represents a white space).</p> <pre>Much__MORE__data__IS__NOT__FOUND</pre> <p>will match</p> <pre>much__more__data__is__not__found</pre> <p>This switch is used only in source comparisons.</p>

COMMANDS

SWITCH	MEANING
/l	The utility compares the files in ASCII mode. It is the default when you compare files that do not have extensions of ".EXE", ".COM", ".SYS", ".OBJ", ".LIB" or ".BIN".
/lb <i>length</i>	Sets the Internal Line Buffer to <i>length</i> lines. The default length of the internal buffer is 100 lines. Files that have more than <i>length</i> consecutive differing lines will abort the comparisons.
/n	The line numbers are displayed on ASCII comparisons.
/t	Tabs are not expanded to spaces. The default is to treat tabs as spaces to 8 column positions.
/w	<p>The utility compresses "whites" (tabs and spaces) during the comparison. Thus, multiple contiguous whites in any line will be considered as a single white space. Note that although FC compresses whites, it does not ignore them. The two exceptions are beginning and ending whites in a line, which are ignored. For example (note that an underscore represents a white space):</p> <p style="padding-left: 40px;">___More__data_to_be_found___</p> <p>will match with</p> <p style="padding-left: 40px;">More_data_to_be_found</p> <p>and with</p> <p style="padding-left: 40px;">_____More_____data_to_be____found_____</p> <p>but will not match with</p> <p style="padding-left: 40px;">___Moredata_to_be_found</p> <p>This switch is used only in source comparisons.</p>

SWITCH	MEANING
/ #	Replace # with the number of lines required to match for the lines within the files to be considered as matching again, after a difference has been found. # can be any number from 1 to 9. If this switch is not specified, the number defaults to 2. This switch is used only in ASCII comparisons.

Note: The default setting for tabs is to convert them into spaces to 8-column positions.

The File Comparison utility reports differences between the two files you specify by displaying the first file name, then the matching line before the differences, followed by the lines that differ between the files, followed by the first line to match in both files. FC then displays the name of the second file followed by the matching line before the lines that are different, followed by the first line that matches. The default for the number of lines that must match before FC recognizes a match is 2. (If you want to change this default, specify this number with the /# switch.) For example:

```

...
...

*****filename1
matching line before differences
difference
1st line to match file2 in file1 after difference

*****filename2
matching line before differences
difference
1st line to match file1 in file2 after differences
*****

...
...
```

If there are too many differences (involving too many lines), the program simply reports that the files are different and stops.

If no matches are found after the first difference is found, FC displays:

resynch failed. Files are too different

and returns to the MS-DOS default drive prompt.

The comparison report is sent to the screen unless you specify output redirection to a file.

FC uses a large amount of memory as buffer (storage) space to hold the source files. If source files are larger than available memory, FC compares only what can be loaded into the buffer space. If no lines match in those portions of the files that have been loaded into the buffer space, FC simply displays the message:

resynch failed. Files are too different

For binary files larger than available memory, FC compares the files piece meal, overlaying the portions in memory with the next portions from disk. All differences are output in the same manner as those files that fit completely in memory.

Examples

Assume these two ASCII files are on disk:

ALPHA.DOC	BETA.DOC
A	A
B	B
C	C
D	G
E	H
F	I
G	J
H	1
I	2
M	P
N	Q
O	R
P	S
Q	T
R	U
S	V
T	4
U	5
V	W
W	X
X	Y
Y	Z
Z	

The following examples show three possible ways of using FC to compare the contents of these two files.

COMMANDS

IF you enter...	THEN...
FC ALPHA.DOC BETA.DOC	<p>FC compares ALPHA.DOC with BETA.DOC and displays the differences on the screen. All the defaults remain intact. The output appears on the screen as follows (the Notes do not appear):</p> <pre> ***** ALPHA.DOC NOTE: ALPHA file C contains CDEFG, D BETA contains CG E F G ***** BETA.DOC C G ***** ***** ALPHA.DOC NOTE: ALPHA file I contains IMNOP, M BETA contains N IJ12P O P ***** BETA.DOC I J 1 2 P ***** ***** ALPHA.DOC NOTE: ALPHA file V contains VW BETA W contains V45W </pre>

IF you enter...	THEN...
	<pre> ***** BETA.DOC V 4 5 W ***** </pre>
FC /4 ALPHA.DOC BETA.DOC > PRN	<p>FC compares ALPHA.DOC with BETA.DOC. 4 lines have to be the same in the comparison, for FC to regard the lines within the file as matching again. The output is redirected to the line printer (PRN).</p> <pre> ***** ALPHA.DOC C D E F G H I M N O P ***** BETA.DOC C G H I J 1 </pre> <p>NOTE: P is the 1st of a string of 4 matches.</p>

COMMANDS

IF you enter...	THEN...
	<p>2 P ***** ***** ALPHA.DOC V W ***** BETA.DOC NOTE: W is the V 1st of a string 4 of 4 matches 5 W *****</p>
<p>FC /b ALPHA.DOC BETA.DOC</p>	<p>the following binary comparison report appears:</p> <p>NOTE: The first field is the relative address bytes from the beginning of the file. The second field is the mismatching byte from ALPHA.DOC. The third field is the mismatching byte from BETA.DOC.</p> <pre> 00000009: 44 47 0000000c: 45 48 0000000f: 46 49 00000012: 47 4a 00000015: 48 31 00000018: 49 32 0000001b: 4d 50 0000001e: 4e 51 00000021: 4f 52 00000024: 50 53 00000027: 51 54 0000002a: 52 55 0000002d: 53 56 00000030: 54 34 00000033: 55 35 00000036: 56 57 00000039: 57 58 0000003c: 58 59 0000003f: 59 5a fc: alpha.doc longer than beta.doc </pre>



FDISK

Sets up the MS-DOS partition for the hard (fixed) disk.

Classification

External, Non-network

FDISK

Characteristics

The FDISK command allows you to set up the MS-DOS partition(s) on the fixed disk. Refer to the "MS-DOS Software Installation Guide" for operational details.



FIND

Searches for a specific string of text in a file or files.

Classification

External

`[d:][path] FIND [/V] [/C] [/N] "string" [pathname] ...`

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where FIND is to be found.
<i>path</i>	Specifies the directory where FIND is to be found.
<i>string</i>	A string of valid characters contained in quotes ("").
<i>pathname</i>	The path of a file to be searched.

Characteristics

FIND displays all lines that contain the specified string from the file or files listed in the command line.

You cannot use wild cards in your file specifications.

If no files are specified, FIND takes the standard input and displays all lines that contain the specified string.

Put in two sets of quotes where the string itself contains quotes.

That is:

FIND "this is a quote ("")" finds the string this is a quote (")

Switches for FIND are:

SWITCH	MEANING
/V	Causes FIND to display all lines not containing the specified string.
/C	Causes FIND to print only the count of lines in each file that contain a match.
/N	Causes each line to be preceded by its relative line number in the file.

Example

IF you enter...	THEN...
FIND "COLOR" BOOK1 BOOK2	FIND displays each line in BOOK1 and BOOK2 (in that order) that contains the string COLOR.
DIR B: FIND/V "DAT"	FIND displays each file name on the disk in drive B that does not contain the string DAT. Note: when using piping in this manner your default drive system diskette must not be write-protected.

COMMANDS

Remarks

If you use the FIND command on BASIC text files, the text must have been saved in ASCII format.

If you use more than one switch with the FIND command, you will get the results summarized in the following table:

COMBINATION	RESULT
all three switches or /V and /C	The count of lines that do not contain the string is displayed.
/C and /N	The count of lines that contain the string is displayed.
/V and /N	The lines not containing the string are displayed, together with their line numbers.

FOR

Allows iterative execution of MS-DOS commands.

Classification

Internal

Interactive entry:

```
FOR %char IN ( item ...) DO [command] %char
```

Batch entry:

```
FOR %%char IN ( item ...) DO [command] %%char
```

Where

SYNTAX ELEMENT	MEANING
<i>char</i>	Any single character other than the digits 0-9.
<i>item</i>	A parameter valid for the command required (separated from another such item by a space).
<i>command</i>	The command you wish to invoke. External commands may optionally be preceded by the drive and/or the path to the directory where the command is to be found.

COMMANDS

Characteristics

Use the *%char* variable in a command line you enter interactively. Use the *%%char* variable in a command line within a batch file. The *%char* or *%%char* variable is assigned the value of each item listed in turn. The command specified is activated for each resulting parameter.

You may include the wild cards * and ? in an item.

Remember to separate each item with a space, and to surround the complete item list by parentheses.

Examples

IF you enter in a batch file...	THEN...
FOR %%f IN (*.ASM) DO MASM %%f	All .ASM files are submitted to the assembler.
FOR %%f IN (report memo address) DO DEL %%f	The files report, memo and address are deleted.

Remarks

A FOR command cannot call another FOR command directly. However a FOR command can call a secondary command processor, which in turn processes another FOR command (see details of COMMAND for a full explanation).

For example:

```
FOR %x IN (1 2) DO COMMAND /C FOR %y IN (%x) DO REM %y
```

produces the output:

```
A> COMMAND /C FOR %y IN (1) DO REM %y
A> REM 1
A>
A> COMMAND /C FOR %y IN (2) DO REM %y
A> REM 2
```

FORMAT

Formats a disk to receive MS-DOS files.

Classification

External, Non-network

*[d:]**[path]* **FORMAT** *drive:* [/1] [/S] [/O] [/V] [/8] [/4]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where FORMAT is to be found.
<i>path</i>	Specifies the directory where FORMAT is to be found.
<i>drive</i>	The name of the drive that contains the disk.

COMMANDS

Characteristics

You must run **FORMAT** on any new diskette you wish to use with **MS-DOS**. Any information already on the diskette is destroyed.

FORMAT places a bootstrap loader directory and file allocation table at the beginning of the diskette. It also checks for any faulty sectors on the diskette.

If you use the **FORMAT** command on a hard disk it will format the logical drive specified. If you are formatting an existing formatted logical drive on a hard disk, **FORMAT** prompts you with the following message:

Enter current Volume Label for drive (x:)

This is a security feature, to prevent you accidentally formatting an existing formatted logical drive. However if that logical drive has no Volume Label, press **ENTER** in response to the message. If the volume label that you enter does not match the label on the hard disk, **FORMAT** displays the message:

Invalid Volume ID Format failure

Otherwise Format continues:

**WARNING, ALL DATA ON NON-REMOVABLE DISK DRIVE x:
WILL BE LOST!**

Proceed with Format (Y/N)?

If you want to format your hard disk, type **Y** and press the **ENTER** key. If you do not want to format your hard disk, type **N** and press the **ENTER** key.

Switch options available with the FORMAT command have the following effect:

SWITCH	MEANING
/1	Formats a diskette single-sided. You would use this option for preparing 180 Kbyte diskettes on a double sided drive (or 160 Kbyte diskette if the /8 option is also specified). This option is not valid on a 1.2 MB, 3 1/2 inch, or hard disk.
/S	Copies the hidden system files and COMMAND.COM to the disk being formatted.
/O	<p>Can only be used in conjunction with the /8 option to leave a place in the directory for the operating system of MS-DOS version 1.1. But the operating system is not placed on the disk.</p> <p>Note: This option causes the FORMAT program to take significantly longer.</p>
/V	Allows you to enter a volume label. The FORMAT command issues a prompt that enables you to enter a unique volume label of up to 11 characters. This label will appear in subsequent directory listings.
/8	Formats diskettes 8 sectors per track instead of the default. 9 sectors per track. Diskettes formatted in this manner are compatible with MS-DOS Ver. 1.XX.
/4	Formats 48 tpi diskettes in High Capacity drives. Note: Diskettes formatted with this switch cannot be reliably used in Normal Capacity drives.

Remarks

Refer to the section on "Disks", in Chapter 1 for charts of "Diskette Capacities" and "Diskette Type Compatibility in Different Capacity Drives".

COMMANDS

Unless you use a switch to specify otherwise, the default format depends on the diskette type and the drive capacity.

Example

IF you enter...	THEN...
FORMAT B: /S	the diskette in drive B is formatted and operating system files are copied onto it.

Remarks

For diskette drives, FORMAT prompts you with a message such as:

**Insert new diskette for drive B:
and strike ENTER when ready**

When you have struck **ENTER** to continue, MS-DOS formats the disk cylinder by cylinder and displays the following information:

Head: x
Cylinder: y

Where the head-value can be 0 or 1, and the cylinder-value increases from 0 to the number of cylinders formatted. When format has finished you will receive a message such as:

Format complete

362496 bytes total disk space
362496 bytes available on disk

Format another (Y/N)?

Press **Y** to format another; **N** to return to MS-DOS.

If you have 3 1/2 inch disk drives the operating system may need

configuring to handle this media. If both the "A:" and "B:" are 3 1/2 inch drives, place the following declarations in your CONFIG.SYS:

DRIVPARM = /D:0 /F:2

DRIVPARM = /D:1 /F:2

(/D:0 indicates drive A:, /D:1 indicates drive B:, /F:2 indicates 720KB capacity.) After changing the CONFIG.SYS file your computer must be re-bootstrapped. See the "MS-DOS Software Installation Guide" for more details.

Error Codes

The following error codes are returned by FORMAT, these can be tested by IF ERRORLEVEL in a batch file:

- 0 Normal completion
- 3 Terminated by user (CTRL BREAK or CTRL C)
- 4 Fatal Error
- 5 N response to hard disk prompt, **Proceed with Format (Y/N)?**



GOTO

Jumps to a specified position in a batch file.

Classification

Internal

GOTO *label*

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>label</i>	<p>A string of characters, the first eight of which are significant (there is no need for quotes around the string). In a batch file, when the GOTO command is executed, the next command executed is on the line following the label. Any line in a batch file can start with a :label. The contents of this line are not displayed by the MS-DOS batch processor. So preceding batch lines by colon (:) is useful for placing comments in a batch file. If the first eight characters of two labels are identical, GOTO that label will cause a jump to the first of the two labels. If no label is found the batch file terminates, with the message:</p> <p>Label not found.</p>

Characteristics

To define a label in a batch file, precede a sequence of characters by a colon (:). Batch processing then ignores the line until it encounters the GOTO command with the label as parameter. It then jumps to the line below the one that contains the label.

Do not enter the colon when using the label as a parameter of GOTO.

Example

IF you execute a .BAT file containing...	THEN...
:foo REM looping... GOTO foo	an infinite number of "looping..." messages are produced.

GRAFTABL

Loads the non-BIOS ASCII characters for graphics modes.

Classification

External

[*d:*][*path*] **GRAFTABL**

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where GRAFTABL is to be found.
<i>path</i>	Specifies the directory where GRAFTABL is to be found.

Characteristics

After the character table is loaded, the following message is displayed:

GRAPHICS CHARACTERS LOADED

The routine is resident and occupies Random Access Memory space. If you try to load GRAFTABL again the following message is displayed.

GRAPHICS CHARACTERS ALREADY LOADED

Remarks

The Random Access Memory space occupied may only be reclaimed by rebooting the system.

Enables graphics currently displayed on the screen, along with any text, to be printed, on a compatible printer, when the **SHIFT PRT SC** keys are pressed.

Classification

External

[d:][path] GRAPHICS [printer-type] [/B][/R][/D{/U | /H}]

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where GRAPHICS is to be found
<i>path</i>	Specifies the directory where GRAPHICS is to be found.

The *printer-type* parameter can be:

printer-type	DESCRIPTION
<i>graphics</i>	Olivetti PR-15B or PR-17B or Industry Standard Graphics Printer or Epson MX/FX 80
<i>color1</i>	Industry Standard Color Printer with black ribbon
<i>color4</i>	Industry Standard Color Printer with RGB (Red, Green, Blue and Black) ribbon 4 colors.
<i>color8</i>	Industry Standard Color Printer with CMY (Cyan, Magenta, Yellow and black) ribbon 8 colors.
<i>jx80</i>	Epson JX-80 (color printer) using the Epson character set.
<i>dm</i>	Olivetti PR-12B (DM285) or PR-14B (DM295) Color Printer. The character set must be IBM International 2.

Remarks

If no *printer-type* parameter is given then *graphics* is assumed.

The GRAPHICS command can only be used with printers that have graphics capabilities.

Switches

Switch options have the following effects:

COMMANDS

SWITCHES	MEANING
/B	Causes the background color to be printed, otherwise the background is suppressed.
/R	Causes black on the screen to be printed black and white on the screen to be printed white. Without the switch the default is to print black as white and white as black.
/D	For a machine with an Enhanced Graphics Color Board (EGC) but no DEB INT 10 filter installed.
/U	400 scanlines (Olivetti High Resolution). This switch cannot be used unless preceded by /D.
/H	200 scanlines (Industry Standard Compatible). This switch cannot be used unless preceded by /D.

Characteristics

This command must be entered to install graphics support necessary to print the screen in graphics modes. SHIFT PRTSC then invokes the printing function. Re-enter the GRAPHICS command with new parameters to reset the existing parameters; graphics support is not reinstalled, only the parameters are changed.

Text modes are printed in the upright position. Graphics Modes are rotated counter-clockwise 90 degrees on the printout page, so the Visual Display Unit's upper right corner appears on the paper's upper left corner.

The GRAPHICS command can only be used with printers that have graphics capabilities.

Remarks

Use INT 5 to print the screen from a program.

Use the GRAPHICS command before entering GW-BASIC if you want to print graphics and text with the GW-BASIC LCOPY command.

For details of printing with an Enhanced Graphics Color Board, see the EGC Board User Guide.

Warning

Do not turn the printer off while printing as this may cause unpredictable effects, and force you to reboot the operating system.

GW BASIC

Enters the MS GW-BASIC interpreter.

Classification

External

[d:][path] **GW BASIC**

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where GWBASIC is to be found.
<i>path</i>	Specifies the directory where GWBASIC is to be found.

COMMANDS

Remarks

For more information on how to initialize GW-BASIC see the "MS GW-BASIC Interpreter under MS-DOS User Guide".

HEXDUMP

Displays the contents of a file, byte by byte, in hexadecimal.

Classification

External

[d:][path] **HEXDUMP** *[drive:]filename*

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where HEXDUMP is to be found.
<i>path</i>	Specifies the directory where HEXDUMP is to be found.
<i>drive</i>	A drive letter, specifying the current directory of the specified drive.
<i>filename</i>	The file whose contents are to be displayed.

Characteristics

Each line of the display shows 16 bytes of information. At the left-hand end of the line the hexadecimal address of the first byte in the line is given. The hexadecimal value of each of the next 16 bytes then follows -two hex digits per byte. The right-hand columns show the ASCII equivalents (if any) of the bytes displayed in that line.

Example

IF you enter...	THEN...
HEXDUMP B:ALPHABET	The contents of the file named "ALPHABET" on the diskette in drive B are displayed in hexadecimal.

HEXDUMP produces a display similar to the following:

Dumping File: B:ALPHABET

```
0000: 20 20 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e " abcdefghijklmn"
0010: 6f 70 71 72 73 74 75 76 77 78 79 7a 30 31 32 33 "opqrstuvwxyz0123"
0020: 34 35 36 37 38 39 2d 5e 40 5b 3b 3a 5d 2c 2e 2f "456789-~@[;:;./"
0030: 2e 7c 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e ".|ABCDEFGHIJKLMN"
0040: 4f 50 51 52 53 54 55 56 57 58 59 5a 5f 21 22 23 "OPQRSTUVWXYZ !"#"
0050: 24 25 26 27 28 29 3d 7e 60 7b 2b 2a 7d 3c 3e 3f "$%&'()=~{+*<>?"
0060: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 " "
0070: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 " "
0080: 0a " "
```

HEXDUMP Complete

Fig. 5-2 HEXDUMP Display

Causes conditional execution of a command in a batch file.

Classification

Internal

IF [NOT] *condition command*

Where

SYNTAX ELEMENT	MEANING
<i>condition</i>	One of the three valid conditions listed below.
<i>command</i>	The command you wish to conditionally execute. If the command is external, it may optionally be preceded by the drive where it is to be found and/or the path leading to the directory where it is to be found.

Characteristics

The specified command is only executed if the condition is true. If it is false the command is ignored. Valid conditions are as follows:

CONDITION	MEANING
EXIST [<i>d:</i>][<i>path</i>] <i>filename</i>	The command is executed only if the specified file exists. on drive <i>d:</i> , and in the directory to which the path leads. The default drive is searched if <i>d:</i> is not specified. The current directory is searched if <i>path</i> is not specified.
<i>string1</i> == <i>string2</i>	The command is executed only if the two strings are identical after parameter substitution. The case of the characters in <i>string1</i> and <i>string2</i> is significant.
ERRORLEVEL <i>number</i>	The command is executed only if the previous program executed had an exit code of the specified number, or higher.
NOT ERRORLEVEL <i>number</i>	The command is executed only if the previous program executed had an exit code of less than the specified number.

COMMANDS

Examples

IF you enter...	THEN...
IF NOT EXIST \SPECIAL\MARKER C:\BIN\CREATE C:\SPECIAL\MARKER	If the file MARKER does not exist the user program CREATE is run to create it.
IF %1 == OLIVETTI ECHO PARAMETER 1 IS OLIVETTI	The computer displays the message only if parameter 1 after substitution is the string "OLIVETTI"
IF NOT ERRORLEVEL 3 LINK	If the error level is under three the linker runs.

JOIN

Joins a disk drive to an empty directory on another drive to produce a single directory structure.

Classification

External, Non-network

Syntax 1 - To join

[d:][path] **JOIN** *connected-drive splice-drive:\splice-directory*

or

Syntax 2 - To deassign a join

JOIN *connected drive:/D*

or

Syntax 3 - To find out which drives are joined

JOIN

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where JOIN is to be found.
<i>path</i>	Specifies the directory where JOIN is to be found
<i>connected-drive</i>	The drive which is to be connected to another drive.
<i>splice-drive</i>	The drive to which reference is to be made.
<i>splice-directory</i>	The connected drive's directory structure is spliced to this directory.
<i>/D</i>	This switch indicates that the connected drive is to be unspliced.

COMMANDS

Characteristics

The JOIN command removes the distinction that physical drives are separately addressable by drive letter. You can refer to all the directories on the joined drives as a single tree structure on one logical drive. After the JOIN command is executed the *connected-drive* becomes invalid.

Remarks

You can only join a *connected-drive* to a *splice-directory* which is a sub-directory of the root of the *splice-drive*. If the *splice-directory* does not exist JOIN will create it. If the *splice-directory* does exist it must be empty of files and sub-directories. Do not JOIN a drive, if the drive being JOINed is part of a substitution (SUBST) or assigned (ASSIGN).

Example: To join drive A: (the *connected drive*) to drive C: (the *splice-drive*). Before the join the "C: Drive Directory Structure" is as shown in the following figure.

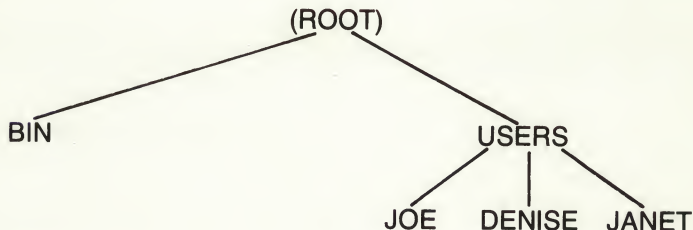


Fig. 5-3 C: Drive Directory Structure

Before the join the "A: Drive Directory Structure" is shown in the following figure.



Fig. 5-4 A: Drive Directory Structure

after the command

`JOIN A: C:\DRIVE-A`

The "Spliced C: Drive Directory Structure" appears to be:

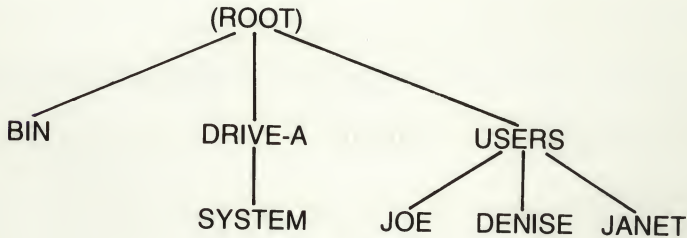


Fig. 5-5 Spliced C: Drive Directory Structure

Notice that in the "Spliced C: Drive Directory structure" that the sub-directory DRIVE-A has been created; the sub-directory SYSTEM of A: has been spliced into C: and the root directory of drive A: has been replaced by the sub-directory DRIVE-A. The whole directory structure of the connected drive is always joined. Drive A: is inaccessible while JOINed. If you try to refer to it the error message is output:

Invalid drive specification

To find out which drives are joined enter **without parameters:**

JOIN

in the above example the following message is output:

A: = > C:\DRIVE-A

To deassign the join in the example enter:

JOIN A: /D

Note: The current directory of the *splice-drive* should always be `\(root)` when JOIN commands are executed. When joining drives, the *connected-drive* should not be the default drive.

Warning

The following commands should not be used while drives are JOINed:

- BACKUP
- DISKCOMP
- DISKCOPY
- FORMAT
- RESTORE



LABEL

Creates, changes or deletes a disk volume label.

Classification

External, Non-network

`[d:][path] LABEL [drive:][volume-label]`

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where LABEL is to be found.
<i>path</i>	Specifies the directory where LABEL is to be found.
<i>drive</i>	The drive containing the disk you wish to LABEL.
<i>volume-label</i>	The name you wish to give the disk. Refer to the section "Parameters" in Chapter 4 for the syntax details of <i>volume-label</i> . From 1 to 11 characters including spaces are allowed.

Characteristics

If you do not specify a volume label, LABEL prompts

Volume in drive X is xxxxxxxxxxxx

Volume label (11 characters, ENTER for none)?

Type the volume label that you want and press the **ENTER** key. See the section "Parameters" in Chapter 4 for the syntax details of *volume-label*. If you want to delete the volume label, just press the **ENTER** key. LABEL prompts with the message:

Delete current volume label (Y/N)?

If you press **Y**, Label deletes the volume label on the disk, otherwise the volume label remains unchanged.

MKDIR

Creates a new directory.

Classification

Internal

Syntax 1

MKDIR [*drive:*] *path*

Syntax 2

MD [*drive:*] *path*

Where

SYNTAX ELEMENT	MEANING
<i>drive</i>	Specifies the drive where the directory is to be created.
<i>path</i>	The path of the directory you want to create.

Characteristics

Use MKDIR to create or add to a hierarchical directory structure on the disk in the default or specified drive.

COMMANDS

You may enter either MKDIR or MD to invoke this command.

Example

IF you enter...	THEN...
MKDIR \USER	The subdirectory USER is created beneath the root directory in the default drive.

MODE

This command enables you to:

- Set the protocol for an RS-232-C port.
- Set the monitor mode.
- Set the printing characteristics or redirect printer output to the communications port.

Classification

External

MODE COM:

Sets the protocol for an RS-232-C port.

[*d*:][*path*] **MODE COM***n*:*baud*[, *parity*[, *databits*[, *stopbits*[, **P**]]]]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where MODE is to be found.
<i>path</i>	Specifies the directory where MODE is to be found.
<i>n</i>	One of the following values: 1 the built-in RS-232-C port 2 the (optional) second RS-232-C port.
<i>baud</i>	The baud rate. This must be one of the following 110, 150, 300, 600, 1200, 2400, 4800 or 9600. Only the first two digits need be specified.

COMMANDS

SYNTAX ELEMENT	MEANING
<i>parity</i>	One of: <ul style="list-style-type: none">- E (even)- O (odd)- N (none) The default is E (even).
<i>databits</i>	The number of data bits. This must be either 7 or 8. The default is 7.
<i>stopbits</i>	Either 1 or 2. If the baud rate is specified as 110, then the default is 2, otherwise the default is 1.
P	Continuous retry on time-out errors.

Characteristics

The baud rate must be specified. All other parameters are optional and will take default values if omitted.

Parameters must be separated by commas.

Use the **P** switch with a serial interface printer. Retry loops resulting from a repeated time-out condition can be broken by pressing **CTRL BREAK**.

Examples

IF you enter...	THEN...
MODE COM1:11,O,8,1	the baud rate is set to 110, odd parity is specified, and the data bits and stop bits are specified as 8 and 1, respectively.
MODE COM1:96	the baud rate is set to 9600. All other parameters take default values.

MODE n

Sets the monitor mode.

Syntax 1

[d:][path] **MODE** n

Syntax 2

[d:] [path] **MODE** [n],m[, T]

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where MODE is to be found.
<i>path</i>	Specifies the directory where MODE is to be found.
<i>n</i>	<p>An argument that may take one of the following values:</p> <p>40 The width of the display is set to 40 characters per line (color monitor only).</p> <p>80 The width of the display is set to 80 characters per line (color monitor only).</p> <p>BW40 Switches the active display controller to that of the color display, sets the display mode to black and white and the display width to 40 characters per line.</p> <p>BW80 Switches the active display controller to that of the color display, sets the display mode to black and white and the display width to 80 characters per line.</p> <p>CO40 Switches the active display controller to that of the color display, sets the display mode to color and the display width to 40 characters per line.</p> <p>CO80 Switches the active display controller to that of the color display, sets the display mode to color and the display width to 80 characters per line.</p>
<i>m</i>	Is R for right shift and L for left shift.
<i>T</i>	Displays a test pattern at the top of the screen.

Characteristics

Use the *m* parameter to shift the display one character (with 40 columns) or two characters (with 80 columns) to the left or right. If you specify the *T* switch, MODE displays a series of numbers at the top of the screen, and prompts you:

Do you see the leftmost 0 (Y/N) ?

if you have entered *R* , or

Do you see the rightmost 9 (Y/N) ?

if you have entered *L* .

In response to either prompt press *N* to shift the display and redisplay the prompt, press *Y* to return to MS-DOS.

Remember, if you omit the *n* parameter, to precede the *m* parameter by a comma.



MODE LPT

Sets the mode of operation for a compatible printer, or redirects the output for any printer to a communications port.

Syntax 1

[d:][path] MODE LPT # : [chars][,spacing] [,P]

Syntax 2

[d:][path] MODE LPT # : = COMn

COMMANDS

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where MODE is to be found.
<i>path</i>	Specifies the directory where MODE is to be found.
<i>#</i>	The printer number (1, 2 or 3).
<i>chars</i>	The number of characters per line.
<i>spacing</i>	Vertical spacing in lines per inch. Its value must be either 6 or 8.
<i>n</i>	The number of the communications port to which printer output is to be redirected.
<i>P</i>	Continuous retry on time-out errors.

Example

IF you enter...	THEN...
MODE LPT1:132,8	The mode of operation of printer 1 is set to 132 characters per line and 8 lines per inch.
MODE LPT1: = COM1:	Output that would normally be sent to the printer is redirected to the first RS-232-C port.

Remarks

The printer must be connected before using this command.

Sends output to the terminal one screen at a time.

Classification

External

`[d:][path] MORE`

Characteristics

Either redirect input through MORE or use MORE as a filter added at the end of a command line. Screen output is displayed one screen at a time, instead of scrolling through its entire contents. At the bottom of each screenful, the prompt **---MORE---** is displayed. Press any key to display the next screenful.

Example

IF you enter...	THEN...
TYPE MYFILE MORE or MORE<MYFILE	The file MYFILE on the default drive is displayed one screen at a time. Note: The diskette on the default drive cannot be write-protected.

PATH

Sets a command search path in the environment.

Classification

Internal

PATH[=] [; | [*pathname* ; *pathname*] ...]

Where

SYNTAX ELEMENT	MEANING
<i>pathname</i>	The path of a directory you wish MS-DOS to search, including optionally a drive letter. Do not include a filename.

Characteristics

PATH tells MS-DOS which directories, and in what order, to look for external commands after it has searched your current directory.

You can specify a single directory path or a number of paths each separated by a semicolon (;), (there must be no spaces in this string). The default is no search path; in this case MS-DOS only searches your current directory.

If you enter **PATH** with no parameter, MS-DOS displays the current search path. If you enter **PATH;** any previously established path is cancelled and only your current directory is searched.

COMMANDS

You only need to set the search path once in any terminal session.

Path only finds executable files: e.g. .COM, .EXE, .BAT files. Path ignores files with any other extension. The paths are searched in the order specified, so place the most frequently accessed directories first.

Example

IF you enter...	THEN...
PATH C:\BIN\USER;C:\BIN\DEV	MS-DOS searches first your current directory, then \BIN\USER, and finally \BIN\DEV on the C: drive.

Remarks

Non-existent directories specified in the PATH variable in the environment are ignored.

PAUSE

Suspends execution of the batch file in which it is contained.

Classification

Internal

PAUSE [*comment*]

Where

SYNTAX ELEMENT	MEANING
<i>comment</i>	A string of up to 121 characters.

Characteristics

When PAUSE is encountered during the execution of a batch file, any comment you have entered shows on the monitor followed by this prompt:

Strike a key when ready...

At this point the batch file is suspended, allowing you to change disks or perform any other necessary action.

To resume batch execution press any key with the exception of **CTRL C** or **CTRL BREAK**.

Press **CTRL C** to cancel processing of the batch file. The following prompt appears:

Abort batch job (Y/N)?

Press **Y** to cancel the batch operation and return to the MS-DOS prompt. Press **N** to return to the previous prompt.

Example

IF you enter...	When the batch file runs...
PAUSE insert target disk in drive B:	the batch job is suspended and "insert target disk in drive B:" displayed.

Queues text files for background printing, while other MS-DOS commands are obeyed.

Classification

External

Syntax 1

The first time PRINT is called

```
[d:][path] PRINT [/D:device] [/B:buffsize] [/U:busyticks] [/M:maxticks]
[/S:timeslice] [/Q:queuesize] [[pathname]...]
```

Syntax 2

Subsequent calls to PRINT

```
[d:][path] PRINT [[ /C//P] [[pathname]...] [/C//P]...]
```

Syntax 3

Subsequent call to terminate PRINT

```
[d:][path] PRINT [/T]
```

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where PRINT is to be found.
<i>path</i>	Specifies the directory where PRINT is to be found.
<i>pathname</i>	The file specification of a file to be printed, optionally preceded by the drive and the path to the directory where the file is to be found.
<i>/D:device</i>	Use to specify the print device. If not used PRINT will ask for a print device.
<i>/B:buffsize</i>	Use to set the internal print buffer size in bytes. The normal size is 512 bytes. Increasing the size may increase performance.
<i>/U:busyticks</i>	Specifies the number of MS-DOS clock ticks that PRINT will wait if the printer is busy. Otherwise PRINT gives up its timeslice. The default is 1 tick.
<i>/M:maxticks</i>	Specifies how many MS-DOS clock ticks print can have to print a file. <i>maxticks</i> can be from 1 to 255 clock ticks (the default is 2).
<i>/S:timeslice</i>	Specifies the time slice value. <i>timeslice</i> can be from 1 to 255 (the default is 8). The lower the value the higher the priority of the print queue.
<i>/Q:queuesize</i>	Specifies the number of files allowed in the print queue. <i>queuesize</i> can be from 1 to 32 (the default is 10).

COMMANDS

Characteristics

You may use global and wildcard characters.

When you run PRINT for the first time in a terminal session, you are prompted as follows:

Name of list device [PRN:]

Type the name of a valid line printer device driver, or simply press **ENTER** to accept the default line printer device PRN:.

The following switches are possible with this command:

SWITCH	MEANING
/T	TERMINATE: this switch cancels all files in the print queue (those waiting to be printed). A message to this effect will be printed.
/C	CANCEL: This switch turns on cancel mode. The preceding filespec and all following filespecs will be suspended in the print queue until /P switch is encountered on the command line.
/P	PRINT: This switch turns on print mode. The preceding filespec and all following filespecs will be added to the print queue until a /C switch is encountered on the command line.

PRINT with no parameters displays the contents of the print queue on your screen without affecting the queue.

Examples

IF you enter...	THEN...
PRINT /T	the print queue is emptied.
PRINT /T *.ASM	the print queue is emptied then all the *.ASM files are queued to the printer.
PRINT TEMP1/C TEMP2 TEMP3	the three files indicated are removed from the print queue.
PRINT TEMP1/C TEMP2/P	TEMP1 is removed from the queue, whereas TEMP2 is added.

Warning

When PRINT is active do not print screen (**SHIFT PRT SCR**) or turn the printer echo on (**CTRL PRT SCR**).

Sets the MS-DOS command prompt.

Classification

Internal

PROMPT `[[meta-character|character]...]`

Where

SYNTAX ELEMENT	MEANING
<i>meta-character</i>	A special character you wish to use to create the MS-DOS prompt, preceded by a \$ sign.
<i>character</i>	A character you wish to appear in your prompt, but this cannot be a \$ or any of the characters described in the following table.

Characteristics

If no argument is entered, the prompt will be set to the default prompt, which is the default drive designation plus the > symbol. You can set the prompt to something different such as the current time, by using the meta-characters indicated below.

The following meta-characters can be used in the prompt command

to specify special prompts. They must all be preceded by a dollar sign (\$) in the prompt command:

SPECIAL CHARACTER	MEANING
\$	The '\$' character.
t	The time.
d	The date.
p	The default drive and the path to the current directory.
v	The version number.
n	The default drive.
g	The '>' character.
l	The '<' character.
b	The ' ' character.
—	A carriage return-linefeed sequence.
s	A space (leading only).
h	A backspace.

COMMANDS

SPECIAL CHARACTER	MEANING
e	ASCII escape (Hexadecimal 1B).
q	The ' = ' character.

Example

IF you enter...	THEN...
PROMPT HELLO	the prompt becomes HELLO
PROMPT \$p\$g	the default drive and the current directory is established as a prompt, E.g. C:BIN >
PROMPT HELLO \$g	the prompt becomes HELLO >
PROMPT \$e[7m\$t\$e[0m	the current time is shown in reverse video. The device ANSI.SYS has to be in CONFIG.SYS, see Appendix B and Appendix C for more details.

RECOVER

Recovers a file or an entire disk containing faulty sectors.

Classification

External, Non-network

[*d:*][*path*] **RECOVER** [*drive:* | *pathname*]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where RECOVER is to be found.
<i>path</i>	Specifies the directory where RECOVER is to be found.
<i>drive</i>	A drive containing a disk with faulty sector(s).
<i>pathname</i>	A file containing faulty sector(s). Optionally preceded by a drive and directory path.

Characteristics

Specify a file name to recover a particular file. RECOVER reads the

COMMANDS

file sector by sector and marks any faulty sectors it finds. MS-DOS will no longer allocate data to such sectors.

Specify a drive name to recover a complete disk. RECOVER reads the contents of the disk sector by sector and marks any faulty sectors it finds.

When you run RECOVER on a disk any directory tree that may be present is destroyed. Files are placed in the root directory and renamed FILE0001.REC, FILE0002.REC and so on. If there is not enough space in your root directory for information on all the files on the disk, the following message is displayed:

Warning - directory full

If this happens copy these recovered files to another disk, delete the FILE*.REC files and run RECOVER again with the same parameters as before. You can neither run RECOVER on a directory nor use a list of files or wild cards.

**REM**

Displays a remark during the execution of a batch file.

Classification

Internal

REM [*remark*]

Where

SYNTAX ELEMENT	MEANING
<i>remark</i>	A string of up to 123 characters.

Characteristics

A remark inserted in a batch file shows on the screen as soon as it is encountered during batch execution. ECHO OFF prevents display of *remark*.

REN

Renames files.

Classification

Internal

REN[AME] *pathname filename*

Where

SYNTAX ELEMENT	MEANING
<i>pathname</i>	The path of the file to be renamed (excluding the drive only if the file is on the default drive).
<i>filename</i>	The new name including any extension you wish to give the file.

COMMANDS

Characteristics

REN changes the name and extension of the file specified in the first parameter to those given in the second parameter. REN cannot be used to move a file from one drive or directory to another; therefore the second parameter must only be a filename.

The wildcard filenames, using * and ?, may be used in either parameter. If wild cards appear in the second parameter, the corresponding characters in the first parameter remain unchanged.

Examples

IF you enter...	THEN...
REN B:PRESENT PAST	the file PRESENT in the current directory in drive B is renamed PAST.
RENAME *.JON *.?A?	any file in the current directory in the default drive with the extension JON has its extension changed to JAN.

Remarks

An attempt to give a file a name already in the file directory results in the following message:

Duplicate file name or file not found

REPLACE

Updates or adds files according to the criteria set by the option switches.

Classification

External

[d:][path] **REPLACE** *[source-drive:][source-path]source-file*
[target-drive:][target-path] [/A][/D][/P][/R][/S][/W]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where REPLACE is to be found.
<i>path</i>	Specifies the directory where REPLACE is to be found.
<i>source-drive</i>	Specifies the drive containing the source files.
<i>source-path</i>	Specifies the directory containing the source files.
<i>source-file</i>	Specifies the source file(s) that are to be added or replaced in the target directory. The files can be specified by wild cards.

COMMANDS

SYNTAX ELEMENT	MEANING
<i>target-drive</i>	Specifies the target drive.
<i>target-path</i>	Specifies the target directory.

Switches for REPLACE are:

SWITCH	MEANING
/A	Causes REPLACE to add new files to the target directory instead of replacing existing ones. You may not use this switch with either the /D or /S switches.
/D	Causes REPLACE to replace files in the target directory only if the source files are newer than the corresponding target files. This switch is incompatible with the /A switch.
/P	Causes REPLACE to prompt you before replacing a target file or adding a source file: Replace (filename)? (Y/N)
/R	This switch causes REPLACE to replace read-only files as well as unprotected files. If you do not specify this switch, any attempt to replace a read-only file causes an error and stops the replace process.

SWITCH	MEANING
/S	This switch causes REPLACE to search all subdirectories of the target directory while it replaces matching files. This switch is incompatible with the /A switch. REPLACE never searches subdirectories in the source path.
/W	This switch causes REPLACE to wait before replacing or adding files. This is useful for changing diskettes.

Characteristics

By default, it replaces files in the target directory with files in the source directory that have the same name. However when you specify the /A switch, REPLACE adds files that exist in the source directory (but NOT in the target directory) to the target directory.

As files are replaced or added, Replace displays the filenames on the screen; then at the conclusion of the replace operation, it displays a summary line:

NNN file(s) added/replaced or No files added/replaced

Note

You cannot use the REPLACE command to update hidden files or system files.

Examples

Replacing Files:

Suppose your hard disk, drive "C:", contains several files of client

COMMANDS

names and phone numbers. To replace these files with the latest version of this file that exists on the disk in drive A, you would enter:

```
REPLACE A:\PHONES.CLI C:\ /S
```

This command replaces every file on drive "C:" that is named PHONES.CLI with the file PHONES.CLI from the root directory on drive "A".

Adding Files:

Suppose you want to add some new printer device drivers to a directory called C:\MSTOOLS, which already contains several printer driver files for a word processor. To do this, you would enter:

```
REPLACE A:*.PRD C:\MSTOOLS /A
```

This command adds any files from the default directory of drive "A:" with an extension of PRD (that do not currently exist in the \MSTOOLS directory on drive "C:") to C:\MSTOOLS.

Error Codes

If Replace encounters an error, it returns one of the following error level codes:

- 1 Command line error
- 2 File Not Found
- 3 Path Not found
- 5 Access Denied
- 8 Insufficient Memory
- 15 Invalid Drive

Other MS-DOS internal extended error number.

You can test for these codes by using the IF ERRORLEVEL command within a batch file.

RESTORE

Restores a number of files from back-up disks. The backup disks must have been created using the BACKUP command.

Classification

External

[d:][path] RESTORE source-drive: [target-drive:] [pathname] [/S] [/P]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where RESTORE is to be found.
<i>path</i>	Specifies the directory where RESTORE is to be found.
<i>source-drive</i>	The drive containing the disk with the backup information to be restored. Typically a floppy disk drive.
<i>target-drive</i>	The drive containing the disk to which the backup information is to be restored. Typically a hard disk drive.

COMMANDS

SYNTAX ELEMENT	MEANING
<i>pathname</i>	The file(s) you wish to restore. If you enter only the hard disk specifier, then all files backed up from the current directory are restored. If you specify a path terminating in a directory name then all files backed-up from that directory will be restored to it. If the path terminates in a file name (or a group of file names specified using wild card characters) then only the specified file(s) will be restored.
/S	Files in all subdirectories, as well as those in the specified (or current) directory will be restored. This includes all levels of subdirectory below the specified (or current) directory.
/P	You will be prompted before restoring files that have been modified since the backup was made, or those files that are read only. This switch is recommended when restoring files backed up from MS-DOS Ver. 2.11 or Ver. 3.10 disks.

Characteristics

Once you have entered the command line you are prompted to insert the backup diskette. It is up to you to ensure that you insert the diskette(s) containing the file(s) you wish to restore in the correct order. If you are unsure as to which diskette(s) contain the files you require, start with the first backup diskette then insert each backup diskette in turn in the order in which they were made. The RESTORE command will prompt you to insert the next diskette.

Note

If you have backed up the root directory of a disk running under MS-DOS Ver. 2.11 or Ver. 3.10, before you restore the root carry out the following procedures at the MS-DOS prompt:

1. C:
2. CD \
3. COPY CON IO.SYS
4. IO.SYS
5. Press F6
6. Press ENTER
7. COPY IO.SYS MSDOS.SYS

You have now created two dummy files called IO.SYS and MSDOS.SYS. These have the same name as the hidden system files used by MS-DOS Ver. 2.11 and Ver. 3.10. When you restore the root directory of the C: drive, use the /P switch. The use of this switch is shown in the following example:

Examples

IF you enter...	THEN...
RESTORE A: C:\ /S/P	<p>All files on a series of backup diskettes in drive "A:" are restored to the hard disk drive "C:" in the same directory structure. Before restoring a file which was changed after it was backed up, RESTORE asks you the following question:</p> <p>filename</p> <p>Warning! The file above was changed after it was backed up.</p> <p>Replace the file (Y/N)?</p> <p>Answer N, if <i>filename</i> is IO.SYS, MSDOS.SYS, COMMAND.COM or any of the other files supplied on your MS-DOS Ver. 3.20 System or Supplementary Diskettes. When you have finished restoring files to the hard disk, the dummy IO.SYS and MSDOS.SYS can be deleted.</p>
RESTORE A: C:*.COM	<p>all files on the backup diskette(s) having the file name extension .COM that were backed up from the current directory are restored into the current directory on the hard disk drive "C:".</p>

The RESTORE command sets the exit code as follows:

- 0 Normal completion
- 1 The specified file(s) was not found
- 3 Command execution terminated by the user
- 4 Command execution terminated due to error

The exit codes can be used by the batch processing IF ERRORLEVEL command.



RMDIR

Removes an empty sub-directory

Classification

Internal

Syntax 1

RMDIR [*drive:*]*path*

Syntax 2

RD [*drive:*]*path*

Where

SYNTAX ELEMENT	MEANING
<i>drive</i>	The letter of the drive containing the directory you wish to remove.
<i>path</i>	The path of the directory you wish to remove.

COMMANDS

Characteristics

RMDIR removes a directory that is empty apart from the . and .. special files. If the directory contains subdirectories or files, these must first be removed by means of the RMDIR or DEL command respectively.

You may either enter RMDIR or RD to invoke this command.

Example

IF you enter...	THEN...
RMDIR C:\BIN\USER\JOE	the specified empty directory is removed, on the C: drive.

SELECT

Formats the target disk and installs MS-DOS, configured for your selected country and keyboard.

Classification

External

*[d:][path] **SELECT** [[s-drive] t-drive:[t-path]] country-code
keyboard-code*

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where SELECT is to be found.
<i>path</i>	Specifies the directory where SELECT is to be found.
<i>s-drive</i>	Specifies the source drive, which can only be A: or B:. If this parameter is not specified the default source drive is A:.
<i>t-drive</i>	Specifies the target drive. If this parameter is not specified the default target drive is B:. The source drive and the target drive must be different.
<i>t-path</i>	Specifies the target directory. If this is not specified, the root directory is the default.
<i>country-code</i>	A three digit number which is the telephonic international country code.
<i>keyboard-code</i>	A two character alpha code indicating your national keyboard.

Note

See the "MS-DOS Software Installation Guide" for more details.

COMMANDS

WARNING

This command **FORMATS** the target drive. Only use it for installing MS-DOS for the first time onto unused disks.

SET

Assigns one string value in the environment to another key string; for use in programs or batch files.

Classification

Internal

SET [*key* = *value*]

Where

SYNTAX ELEMENT	MEANING
<i>key</i>	The key string you wish to assign a value to.
<i>value</i>	The string you wish to assign to the key string. Note: leading spaces are significant.

Characteristics

Use SET to assign a value to a standard parameter included in an application program. The value remains operative during a working session until another SET command is issued.

When the SET key command is executed, with a key assignment, it inserts the entire string into a part of memory reserved for "environment" strings. If the name already exists in the environment, it is replaced by the new string. If you enter the SET command with only the first string, the associated string name is removed from the environment. If you enter SET with no options, MS-DOS displays the current environment settings.

A program can get a listing of all the environment values that have been set by examining its environment. (A pointer to the environment is passed in the Program Segment Prefix). Refer to the "MS-DOS System Programmer Guide" for more information.

You can also use the SET command with batch files. Instead of passing string values to a batch file by means of replaceable parameters (see Chapter 4) in the command line, you can use SET to assign string values to string keys. Within the batch file the form of the `key` must be as follows:

`%key%`

That is, the string must be preceded and followed by a percent sign (whereas replaceable parameters are only preceded by a percent sign).

Use of the SET command with no parameters causes all the current SET assignments to be displayed on the screen.

Example

IF you enter...	THEN...
SET TTY = VT52	the TTY value is set to VT52, in all batch files up on execution %TTY% is replaced by VT52.

COMMANDS

Remarks

The case of the key is converted to uppercase. The case of the value is left in the case input. Be careful, because some programs require values in uppercase, these values must be input in uppercase..

For example:

IF you enter...	THEN...
SET tty = vt52	in the environment tty -> TTY
SET	PATH = COMSPEC = A:\COMMAND.COM TTY = VT52

If the program expects VT52 in uppercase vt52 in lowercase will not be recognized.

SHARE

Installs network file and record locking. It also installs a resident facility which checks for diskette removal during reading and writing to diskettes.

Classification

External

[d:][path] **SHARE** [/F: *memory-space*] [/L: *locks*]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where SHARE is to be found.
<i>path</i>	Specifies the directory where SHARE is to be found.
<i>/F:memory-space</i>	When this switch is used, the memory-space parameter indicates how much memory space is reserved for recording file sharing information. Each open file should be allocated space for its pathname plus eleven bytes; so the space allocated should be between 32 bytes and 74 bytes per file. 74 bytes will allow for a full-length pathname of 63 characters. The default memory-space for file-sharing information is 2048 bytes.
<i>/L:locks</i>	When this switch is used, memory space is allocated for the maximum number of locks it is possible to apply to a file. The default value for the number of locks it is possible to apply is 20.

Characteristics


If used it should be included in the system disk's AUTOEXEC.BAT file. Once the command has been called the support utility becomes resident. It takes about 5K bytes of memory (with the default switch settings).

After SHARE is resident, all read and write requests are checked for lock violations. See the "MS-DOS System Programmer Guide" for details on file locking and unlocking.

After SHARE is resident, removal of a diskette during a read or write, causes a check of the replaced diskette's volume label. If the volume label has changed, MS-DOS issues one of the following error messages:

Invalid disk change reading/writing drive x:

SHIFT



Allows access to more than 9 replaceable parameters in batch processing.

Classification

Internal

SHIFT

Characteristics

Usually, batch files are limited to handling 10 parameters, %0 through %9. Initially %0 contains the name of the batch file. Initially %1 to %9 contain the values of the first nine replaceable parameters. To allow access to more than nine replaceable parameters, use SHIFT to alter the numbering of your command line parameters. For example:

calling a batch file SUPER.BAT with the 12 replaceable parameters:

SUPER p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12

initially: %0 = SUPER
 %1 = p1
 .
 .
 %9 = p9

to access the other parameters SHIFT is used within the batch file one SHIFT will result in:

 %0 = p1
 %9 = p10

Successive SHIFT calls will result in:

 %9 = p11

and another SHIFT call will result in:

 %9 = p12

so finally:

 %0 = p3
 %1 = p4
 %2 = p5
 %3 = p6
 %4 = p7
 %5 = p8
 %6 = p9
 %7 = p10
 %8 = p11
 %9 = p12

So if you have entered more than nine replaceable parameters on the command line, those that appear after the ninth (%9) will be shifted one at a time into %9 by successive shifts. You can then refer to these parameters in your batch file.

A filter which sorts data alpha-numerically in forward or reverse order.

Classification

External

[d:][path] SORT [pathname] [/R] [/ + number]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where SORT is to be found.
<i>path</i>	Specifies the directory where SORT is to be found.
<i>pathname</i>	The path of the file to be sorted.
<i>/R</i>	This switch indicates a reverse ASCII sort.
<i>/ + number</i>	The column number from which to begin the sort.

Characteristics

Sort is a filter which works on text lines, the case of the characters is ignored.

Sort takes place in ASCII order unless you specify the **/R** switch, in which case a sort in reverse ASCII sequence is done. It starts with the first column of input unless you specify otherwise using the **/+number** switch.

If you do not specify a file, SORT takes the standard input and outputs to the screen, unless you specify otherwise with the redirection symbols ">" and "<" or the pipe symbol "|".

Note

The maximum size of file that can be sorted is 63 KB.

Examples

IF you enter...	THEN...
<code>SORT /R <UNSORT.TXT > SORT.TXT</code>	the file UNSORT.TXT is sorted in reverse order and the result placed in the file SORT.TXT.
<code>DIRSORT /+ 14 MORE</code>	the directory listing produced by the DIR command is sorted starting with the fourteenth column (the column that contains the file size), and output a screen at a time.

COMMANDS

Collating Sequence

So as to be able to sort National Character Sets the following character mappings are effected:

ORIGINAL CHARACTER	MAPPED CHARACTER
Ç	C
Ü	U
é	E
â	A
ä	A
à	A
å	A
ç	C
ê	E
ë	E
è	E
ï	I
î	I
ì	I
Ä	A
Å	A
É	E
æ	A
Æ	A

ORIGINAL CHARACTER	MAPPED CHARACTER
ô	O
ö	O
ò	O
û	U
ù	U
ÿ	Y
ö	O
ü	U
¢	\$
£	\$
¥	\$
Pt	t
f	\$
á	A
í	I
ó	O
ú	U
ñ	N
Ñ	N
<u>a</u>	<u>a</u>
<u>o</u>	<u>o</u>
¿	?
Г	П
Г	П
1/2	1/2
1/4	1/4
ì	ì
«	”
»	”
β	S

Substitute a dummy drive specifier for a pathname.

Classification

External, Non-network

Syntax 1

[d:][path] SUBST *dummy-drive: pathname*

or

Syntax 2

[d:][path] SUBST *dummy-drive: /D*

or

Syntax 3

[d:][path] SUBST

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where SUBST is to be found.
<i>path</i>	Specifies the directory where SUBST is to be found.
<i>dummy-drive</i>	The dummy drive that is to be used to refer to pathname.
<i>pathname</i>	The drive and/or directory path to which the dummy drive refers.
<i>/D</i>	A switch which indicates that the specified dummy drive substitution should be deleted.

Characteristics

Enter SUBST with no parameters to display the current substitution, for example the resulting output could be:

M = > C:\USR\MIKE

The dummy-drive must be within the range of drives recognized by the system. If you use real drive letters for the dummy-drive, you will not be able to use the real drive. Do not use the default drive as a dummy-drive. It is recommended that you do not use real drives. Increase the availability of drive letters by setting in CONFIG.SYS *LAST DRIVE = dummy-drive* or greater (see Appendix C).

Suppose you have a configuration with a hard disk "C:" and two floppy disk drives ("A:" and "B:"), and are using a dummy drive = M. In CONFIG.SYS set *LASTDRIVE = M*. Call the command

COMMANDS

SUBST M: C:\USR\MIKE

where MIKE is a directory. You would now be able to refer to the directory by using the alias "M:" instead of the path C:\USR\MIKE.

After substitution the command

DIR M:

could produce the following display.

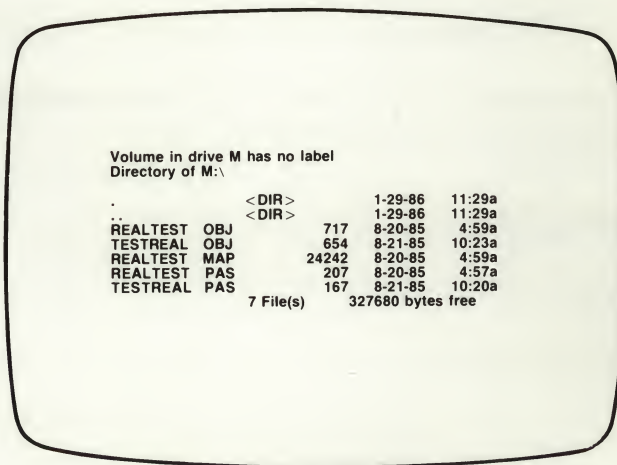


Fig. 5-6 Substituted Directory Display

Substituting is particularly useful for programs that do not recognise paths, or you can use a letter as shorthand for a long path.

To undo a substitution use SYNTAX 2.

In the above example:

SUBST M: /D

will undo the substitution.

Remarks

Never use the SUBST command and then use the following commands on the dummy drive or unpredictable results and/or error messages will occur.

ASSIGN
BACKUP
DISKCOMP
DISKCOPY
FDISK
FORMAT
JOIN
LABEL
PRINT
RESTORE

Pay attention to the substitutions in effect when using the following commands.

CHDIR
MDIR
MDIR
PATH

Updates the specified disk with the hidden system files, which come from the default drive.

Classification

External, Non-network

[*d*:][*path*] **SYS** *drive*:

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where SYS is to be found.
<i>path</i>	Specifies the directory where SYS is to be found.
<i>drive</i>	The drive that contains the target disk.

Characteristics

Bootstrap off the system disk containing the SYS.COM file. With this disk in the default drive, enter the SYS command.

The target disk must either be formatted, but without files in the root directory, or have been formatted with the /S parameter to contain

previous versions of the system files. If this is not the case one of the following messages appears:

SYS cannot install MS-DOS on this disk

or

Not enough room for MS-DOS on this disk

Remarks

The file COMMAND.COM is not transferred.

The hidden files will not appear in any directory listing.

TIME

Displays or set the system time.

Classification

Internal

TIME [*hh[:mm]*]

Where

SYNTAX ELEMENT	MEANING
<i>hh</i>	Hours (0-23).
<i>mm</i>	Minutes (0-59).

COMMANDS

Characteristics

If you type one or more value(s) and then press **ENTER**, **TIME** sets the remaining value(s) to zero. For example, if you enter "8" the time is set to 8:00:00.00

If you leave out the complete parameter, **TIME** prompts you as in the following example:

Current time is 16:36:00.00

Enter new time:

You can then enter a new time in the correct format. To accept the current time simply press **ENTER**.

Example

IF you enter...	THEN...
TIME 8:30	half past eight in the morning is set as the current time.

Remarks

Note that the format of the time output varies depending on the **COUNTRY** configuration in **CONFIG.SYS** (see Appendix C).



TREE

Displays all the directories and paths on the specified drive. It also has an option to list the files in each directory.

Classification

External

`[d:][path] TREE [drive:] [/F]`

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where TREE is to be found.
<i>path</i>	Specifies the directory where TREE is to be found.
<i>drive</i>	The drive whose directory structure is to be examined.
<i>/F</i>	The files contained in each directory are also to be listed.

COMMANDS

Characteristics

For every directory on the specified drive the TREE command will display:

- The directory path starting from the root directory of the drive.
- The subdirectories within the directory.
- The files in the directory (only if /F is specified).

Example

IF you enter...	THEN...
TREE A: /F	the path to each directory on drive A is displayed along with the subdirectories and files defined within each directory.

TYPE

Displays the contents of the specified file on the video screen.

Classification

Internal

TYPE *pathname*

Where

SYNTAX ELEMENT	MEANING
<i>pathname</i>	The path to the file to be displayed, including the filename. Filename can not be wildcarded.

Characteristics

Use this command to examine a file without modifying it. Press **CTRL NUMLOCK** (or **CTRL S**) to suspend output, press any key to recommence typing. Press **CTRL BREAK** (or **CTRL C**) to terminate output. Press **CTRL PRtSC** (or **CTRL P**) to turn the printer on, press **CTRL PRtSC** again to turn the printer off.

The complete contents of the file, including any non-alphabetic and non-numeric characters, appear on the screen. As such, the file may appear unreadable.

Tab characters are expanded on the screen to tab stops every eighth column.

Remarks

It is advised that you can only type (and print) text files.

COMMANDS

VER

Displays the MS-DOS version number.

Classification

Internal

VER

Characteristics

This command displays on your screen the version number of the MS-DOS system you are using.

VERIFY

Sets an internal switch which causes disk writes to be verified.

Classification

Internal

VERIFY [ON|OFF]

Characteristics

This command has the same purpose as the /V switch in the COPY

command. If you want to verify that all files are written correctly to disk, you can use the VERIFY command to tell MS-DOS to verify that your files are intact (no bad sectors, for example). MS-DOS will perform a VERIFY each time you write data to a disk. You will receive an error message only if MS-DOS was unable to successfully write your data to disk.

VERIFY ON remains in effect until you change it in a program (by a SET VERIFY system call), or until you issue a VERIFY OFF command to MS-DOS.

If you want to know what the current setting of VERIFY is, enter VERIFY with no options.

VOL

Displays the volume label of the disk in the specified or default drive.

Classification

Internal

VOL [*drive:*]

Where

SYNTAX ELEMENT	MEANING
<i>drive</i>	The drive that contains the disk to be examined.

COMMANDS

Characteristics

If the disk does not have a volume label, VOL displays the following message:

Volume in drive x has no label

XCOPY

Copies files and directories, including lower level directories if they exist.

Classification

External

**[d:][path] XCOPY [source-drive:][source-path]source-filename
[target-drive:][target-path][target-filename] [/A][/D:mm-dd-yy]
[/E][/M][/P][/S][/V][/W]**

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where XCOPY is to be found.
<i>path</i>	Specifies the directory where XCOPY is to be found.
<i>source-drive</i>	Specifies the drive containing the source files.
<i>source-path</i>	Specifies the directory containing the source files.
<i>source-file</i>	Specifies the source file(s) that are to be copied. The files can be specified by wild cards.
<i>target-drive</i>	Specifies the target drive.
<i>target-path</i>	Specifies the target directory.
<i>target-filename</i>	Specifies the target file(s). The files can be specified by wild cards.

COMMANDS

Switches for XCOPY are:

SWITCH	MEANING
/A	Causes XCOPY to copy source files that have their archive bit set. It does not modify the archive bit of the source file. Refer to the ATTRIB command for information on how to set the archive attribute.
/D	Causes XCOPY to copy source files that have been modified on or after the date specified by <i>mm-dd-yy</i> . Note that the date format may vary depending on the country code that you are using.
/E	Causes XCOPY to copy empty subdirectories. You must use this switch with the /S switch.
/M	This switch is similar to the /A switch since it copies archived files only; however, it turns off the archive bit in the source file. Refer to the ATTRIB command for information on how to set the archive attribute.
/P	Causes XCOPY to prompt you before copying each file: <i>target-filename (Y/N)?</i>
/S	Causes XCOPY to recursively copy lower level subdirectories and their files. Empty subdirectories are not copied unless the /E switch is used with this /S switch. If you omit the /S switch, XCOPY works only within a single directory.
/V	Causes XCOPY to verify each file as it is written to the target to make sure that the target files are identical to the source files.
/W	This switch causes XCOPY to wait before it starts copying files. This is useful for changing diskettes.

Characteristics

When this command is used without parameters, it is the equivalent of simple or wildcarded file copy. When the /A or /M switch is used the command is useful for backing up disks, as an alternative to the BACKUP command. When the /S switch (and optionally the /E switch) is used with the source-directory being root (\), this command can copy whole disks; in contrast to COPY, which can only copy files, one directory at a time. When the /S switch (and optionally the /E switch) is used with the source-directory being a subdirectory, a directory sub-tree is copied.

Examples

```
ATTRIB +A C:\BIN
```

```
XCOPY C:\BIN A: /A
```

The above example copies the whole of the BIN directory on the hard disk to a diskette in drive A:; setting the archive bit in each file with ATTRIB prepares for the use of the /A switch. If there are too many files to fit onto the diskette in A:, use the /M switch instead as in the following example:

```
ATTRIB +A C:\BIN
```

```
XCOPY C:\BIN A: /M
```

When the target disk becomes full, XCOPY finishes and the following message is displayed:

Disk Full

Put another diskette into the A: drive and repeat:

```
XCOPY C:\BIN A: /M
```

Those files that had been copied onto the original diskette had had their archive bit turned off, so they will not be copied onto the second diskette. If the "Disk Full" message is displayed again, repeat the copy operation with fresh target diskettes, until no message is received. However, this technique will not cope with files that are too big to fit on the target diskette, in this case use the BACKUP command instead.

COMMANDS

The following examples illustrate recursive copying. The source drives for both examples have their directory structure illustrated in the following figure:

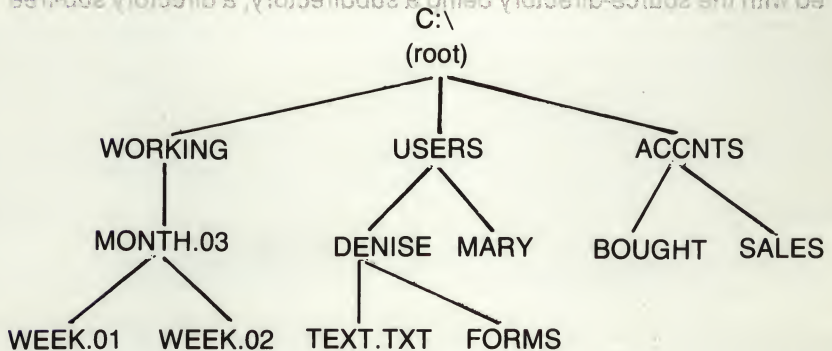


Fig. 5-7 Source Drive Directory Structure

If the target drive directory structure before copying is as illustrated in the following figure:

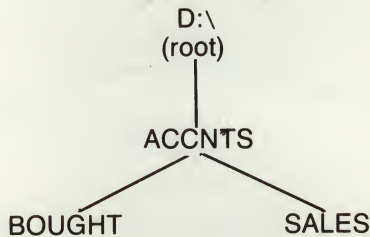


Fig. 5-8 Target Drive Directory Structure Before Copying

Example: The source and the target subdirectory trees have different structures:

```
XCOPY C:\USERS\*.* D: /S
```

The result on the target is to create subdirectories with the same names as those in the source directory tree being copied. All the files that exist in the source subdirectory and its subtree on "C:" will be copied into the directories of the same names on "D:". Note that empty subdirectories will not be created on the target unless the "/E" switch was specified. The resultant directory structure is illustrated in the following figure:

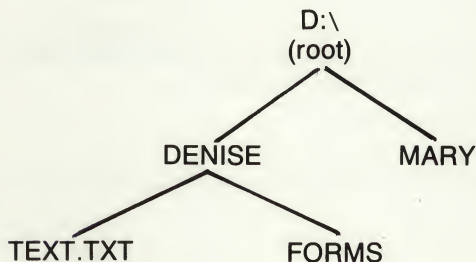


Fig. 5-9 Target Drive Directory Structure After Copying

Example: The source and target drives have the same structure:

```
XCOPY C:\ACCNTS\*. * D:\ACCNTS /S
```

The result on the target is to leave the Target Drive Directory Structure the same as before copying. All the files that exist in the source subdirectory and its subtree on C: will be copied into the directories of the same names on "D:".

Example: Copying From one directory to another:

```
XCOPY C:\WORKING D:\ARCHIVE /S/M
```

As the directory "ARCHIVE" does not exist on the target, it will be created. However "XCOPY" does not know whether the name "ARCHIVE" is intended to be a file or a directory. Therefore you will be asked:

**Does ARCHIVE specify a file name
or directory name on the target
(F = file, D = directory) ?**

Answer **D** as "ARCHIVE" is intended to be a directory.

The subdirectory tree of the directory "WORKING" will be copied and the subdirectories will retain the same names as those in the source directory. All files in WORKING and its subdirectories will be copied to the equivalent target directories. The resultant directory structure is illustrated in the following figure:

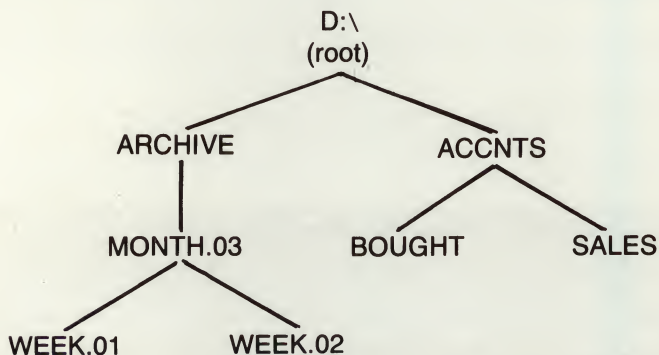


Fig. 5-10 The Target Directory Structure After Copying

Remarks

XCOPY does not provide for copying to or from reserved device names such as CON: or AUX:.

COMMANDS

XCOPY does not provide for pathnames of longer than 63 characters. For deep trees use SUBST to substitute a logical drive letter for long pathnames.

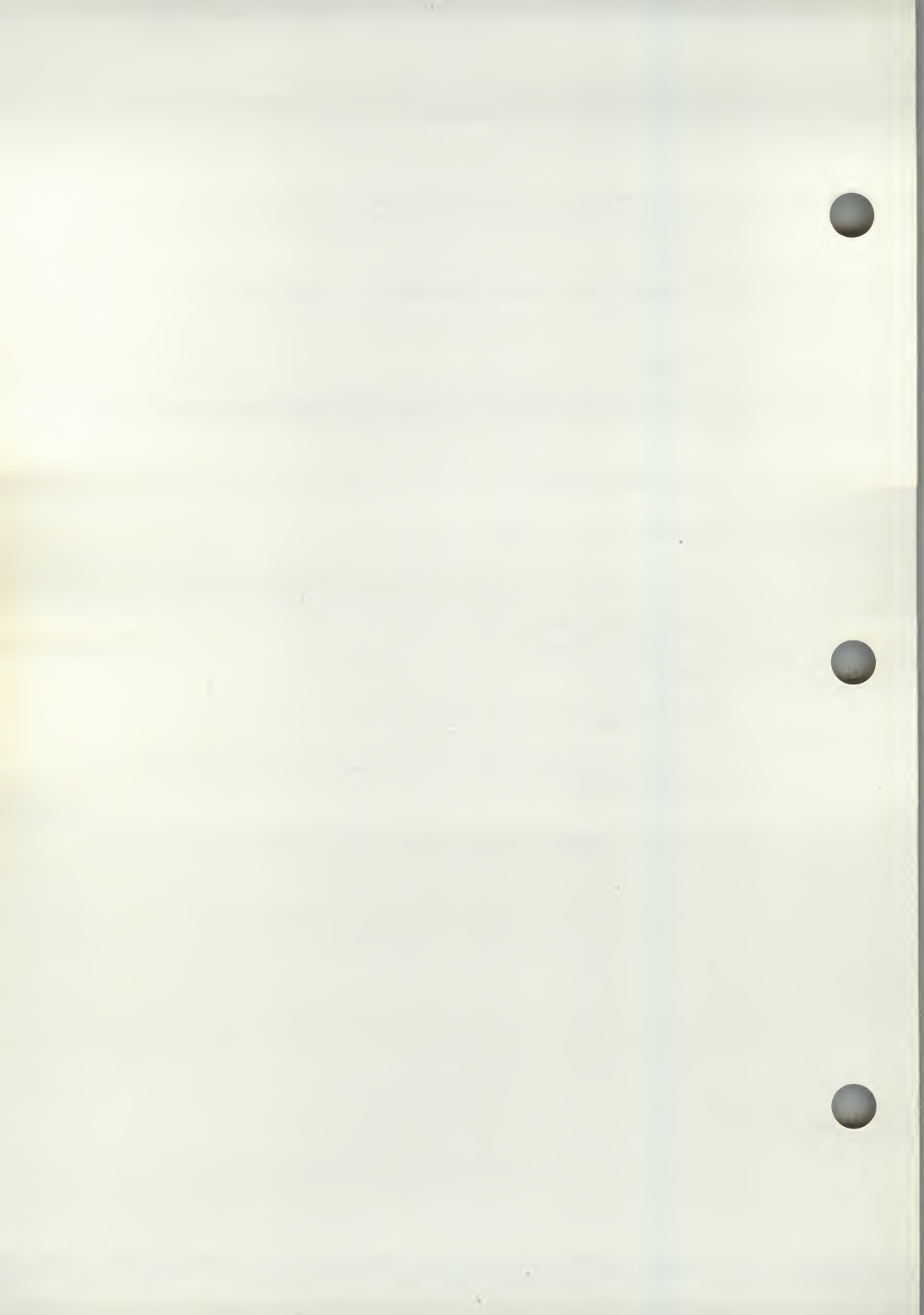
XCOPY cannot copy to or from hidden or protected files.

Error Codes

If XCOPY encounters an error, it returns one of the following error level codes:

- 0 Copy without error
- 1 No files found to copy
- 2 CTRL C entered by user to terminate XCOPY
- 4 Initialization error:
 - Insufficient memory
 - Invalid drive
 - Invalid syntax
 - File not found, or path not found.
- 5 Int 24 error occurred. The user aborted from INT 24 error reading or writing disk.

You can test for these codes in a batch file by using IF ERRORLEVEL.



**6. VIDEO FILE EDITOR
(EDIT)**

ABOUT THIS CHAPTER

This chapter tells you how to use the Video File Editor (EDIT)

CONTENTS

INTRODUCTION	6-1
HOW TO INVOKE THE VIDEO FILE EDITOR	6-1
THE DISPLAY	6-3
THE KEYBOARD	6-4
EDIT	6-7
GENERAL EDITING FUNCTION KEYS	6-8
WINDOW MOVING FUNCTION KEYS	6-16
EXITING AND SAVING FUNCTION KEYS	6-18
COMMANDS AND SEARCHING	6-19
STRING SEARCHES	6-19
COMMANDS	6-21
GOTO	6-21
DELETE	6-22
FILE	6-22

INTRODUCTION

The Video File Editor enables you to create and edit files of text. A text file is a file of records containing printable ASCII characters, and each record is separated from the next by a carriage-return/line-feed pair.

The Video File Editor displays a 21-line "window" within which you can perform editing functions via the keyboard. A subset of these functions enables you to move the window to access any part of the file.

In addition to the functions mentioned above the Video File Editor can also perform an extensive set of line editing and cursor moving functions and can operate in overstrike, insert text or command mode. The latter enables a subset of high level commands.

Each text line in a newly created file can contain up to 80 characters. Existing files created by means other than the Video File Editor can be edited with it, even if its lines extend beyond 80 characters. However, all characters after column 80 are overstruck on column 80.

HOW TO INVOKE THE VIDEO FILE EDITOR

**EDIT**

The EDIT command is used to enter the Video File Editor.

[d:][path] EDIT [/B]/[T]/[R] *pathname*

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	specifies the drive where EDIT is to be found.
<i>path</i>	specifies the directory where EDIT is to be found.
<i>/B</i>	A backup of the file is to be made when the Video File Editor is entered. This backup is named filename.bak where filename is the same as that specified in the command line.
<i>/T</i>	The size of the file will be minimized by automatically replacing multiple spaces with TAB characters wherever possible.
<i>/R</i>	The read-only option and is used when you only wish to examine the contents of the file. This protects the file from accidental damage while examining it.
<i>pathname</i>	The path to the file to be edited.

Characteristics

If the file does not already exist the prompt "OK to Create?" appears on the screen, to which you must type "Y" to create the file.

The Video File Editor is initially in "overstrike" mode. That is, you can enter text and overwrite whatever is already written on the file. The methods of entry into other modes of operation are described later.

THE DISPLAY

Once the Video File Editor has been invoked the monitor shows a display such as:

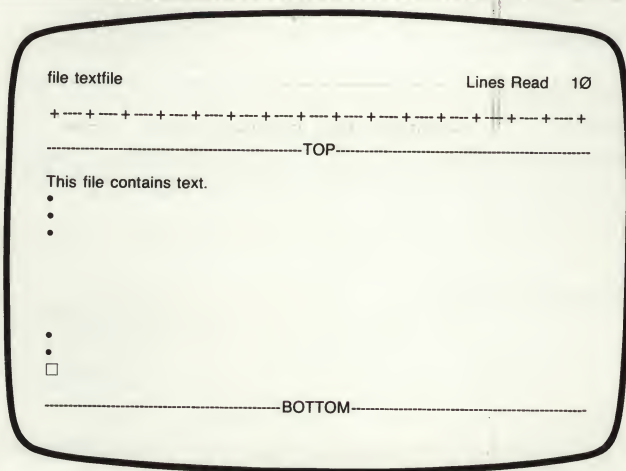


Fig. 6-1 Video File Editor Screen Layout

Line 1 indicates the file name and the current message.

Line 2 is used for high level commands and search strings and is therefore only used when in command mode. Refer to the section entitled "Commands and Searching" for details.

Line 3 shows the tab stop settings (4 character positions per tab).

Lines 4 to 24 contain the text window.

Line 25 is not used.

On entering the Video File Editor the beginning and end of the file are marked by two display lines containing the words TOP and BOTTOM, respectively. The former, known as the TOP bar, always appears immediately before the first line of text in the file. And the BOTTOM bar always appears immediately after the last line of text. They are not actual lines of text and are there merely as markers. The cursor is initially positioned on the TOP bar.

The cursor changes shape when switching between certain modes of editing. It is represented here as underline.

Note that the screen mode for the Video File Editor is 80x25 lines, even if it is invoked from a terminal set to 40x25 lines.

THE KEYBOARD

The keyboard functions in a different manner once the Video File Editor has been invoked. This provides the means by which the required editing functions are entered. The following tables show for each function key, the function name and the key-stroke combination that executes that function.

Using the Numeric Keypad

KEY-STROKE	FUNCTION KEY NAME
HOME	TOP
END	BOTTOM
PGUP	FULL SCREEN UP
PGDN	FULL SCREEN DOWN
←	CURSOR LEFT
→	CURSOR RIGHT
↑	CURSOR UP

VIDEO FILE EDITOR

KEY-STROKE	FUNCTION KEY NAME
↓	CURSOR DOWN

Using the Function Keys

Note that these function keys are summarized on a template supplied with your system. Keep this template by your keyboard, for quick reference during working sessions.

KEY-STROKE	FUNCTION KEY NAME
F1	COMMAND MODE
SHIFT F1	ABORT
F2	RESTORE LINES
SHIFT F2	DELETE LINE
F3	JOIN LINES
SHIFT F3	SPLIT LINE
F4	END OF LINE
SHIFT F4	START OF LINE

KEY-STROKE	FUNCTION KEY NAME
F5	SAVE
SHIFT F5	SAVE AND EXIT
F6	NEXT LINE
SHIFT F6	ERASE TO END
F7	GOTO MARK
SHIFT F7	INSERT MARK
F8	SEARCH DOWN
SHIFT F8	SEARCH UP
F9	LINE DOWN
SHIFT F9	LINE UP
F10	HALF SCREEN DOWN
SHIFT F10	HALF SCREEN UP

VIDEO FILE EDITOR

Using Control Keys

KEY STROKE	KEY FUNCTION NAME
CTRL H	BACKSPACE
CTRL I	TAB
CTRL K	ERASE TO END OF LINE
CTRL L	REFRESH
CTRL R	RECALL LINE
← or BS	BACKSPACE
INS	INSERT MODE
DEL	DELETE CHAR
ESC	ESCAPE
← →	TAB
SHIFT ← →	REVERSE TAB
ENTER ↵	INSERT LINE or EXECUTE COMMAND


GENERAL EDITING FUNCTION KEYS

The keys whose functions are described below perform general editing functions such as moving the cursor and inserting and deleting text.

CLASS	FUNCTION KEY	MEANING
to move the cursor	↑ (CURSOR UP)	Moves the cursor one line up the screen but keeps the same position within the line. If the cursor was on the second line of the window then the window is moved one line up the file and the cursor remains on the second line.
	↓ (CURSOR DOWN)	Moves the cursor one line down the screen but keeps the same position within the line. If the cursor was on the penultimate line of the window, it stays there and the window is moved down one line.
	← (CURSOR LEFT)	Moves the cursor one character position to the left within the same line.

VIDEO FILE EDITOR

CLASS	FUNCTION KEY	MEANING
to move the cursor	→ (CURSOR RIGHT)	Moves the cursor one character position to the right within the same line.
	← → or CTRL I (TAB)	Moves the cursor one tab position (four characters) to the right.
	SHIFT ← → (REVERSE TAB)	Moves the cursor one tab position (four characters) to the left.
	SHIFT F4 (START OF LINE)	Moves the cursor to the start of the current line.
	F4 (END OF LINE)	Moves the cursor to the character position immediately following the last non-space character in the current line.
to insert text	INS (INSERT MODE)	Is entered from overstrike mode. The cursor changes its shape to show that a new mode has been entered. Any character which is subsequently entered is inserted immediately before the cursor position, and the remainder of the text in the line and the cursor are moved one character position to the right. Any character that was in the last character position in the line is discarded. Striking the INSERT MODE key a second time returns the Video File Editor to overstrike mode and the original cursor is restored.

CLASS	FUNCTION KEY	MEANING
to insert text	ENTER  (INSERT LINE)	<p>Inserts a blank line immediately after the current line and places the cursor at the beginning of that line. Subsequent text is pushed one line down the screen.</p> <p>If the cursor was already on the bottom line of the screen then the window is moved one line down the file and the blank line is inserted on the last line of the window.</p>
to delete text	← (or BS) or CTRL H (BACKSPACE)	<p>Moves the cursor one character position to the left and deletes the character under the cursor. Subsequent characters in the line do not move. The deleted characters are replaced with spaces.</p> <p>This function is usually used for correcting typing errors when entering new text.</p>
	DEL (DELETE CHAR)	Deletes the character under the cursor and shifts the subsequent characters in the line one position to the left.
	SHIFT F6 or CTRL K (ERASE TO END)	Deletes the contents of the current line from the current cursor position to the end of the line.

VIDEO FILE EDITOR

CLASS	FUNCTION KEY	MEANING
to delete text	SHIFT F2 (DELETE LINE)	Deletes the current line and moves subsequent text one line up the screen. The position of the cursor is not changed, it remains in the same column position. The deleted line of text is placed in a holding area called the restore buffer. This action overwrites the previous contents of the restore buffer except where DELETE LINE functions immediately follow each other, in which case subsequent deleted lines are appended to the buffer. This enables you to move a block of text from the file into the buffer, from where it can be re-inserted into the same or another file using the RESTORE LINES function.
to restore text	CTRL R (RECALL LINE)	Restores the contents of the current line to its original state. The contents restored are those that existed before the cursor was moved to this line. Once the cursor is moved off a particular line the old contents of that line cannot be recalled using this function.
	F2 (RESTORE LINES)	Inserts the contents of the restore buffer into the file starting at the the line below the current cursor position. The cursor is moved to the start of the inserted line(s). The restore buffer itself is not changed. This function is used in conjunction with the DELETE LINE function to move and/or copy blocks of text.

CLASS	FUNCTION KEY	MEANING
to split and join lines of text	SHIFT F3 (SPLIT LINE)	Divides the current line into two by moving all text under and to the right of the cursor onto the next line. The cursor does not move. Text on subsequent lines is shifted one line down the screen.
	F3 (JOIN LINES)	Combines two lines into one. The text on the subsequent line is placed immediately after the last non-space character on the current line. The cursor does not move. If the current line cannot accommodate the entire text of the next line then only that amount which fits is moved and the remaining text stays on the same line but is moved to the left hand edge of the screen.
to insert a marker	SHIFT F7 (INSERT MARKER)	Causes a marker to be inserted in the text immediately following the current line. The marker is a dotted line containing the text "MARK". If the MARK line was previously located somewhere else in the text it is moved from where it was to the new position. Note that this is not an actual line of text and will never be written to the file. Its placement is therefore only significant during the current editing session. It is used in conjunction with the GOTO MARK function as a place marker (for details see the section entitled "Window Moving Function Keys"), and in conjunction with the high-level command DELETE (see the section entitled "Commands and Searching").

VIDEO FILE EDITOR

CLASS	FUNCTION KEY	MEANING
to enter control characters	ESC (ESCAPE)	<p>The Video File Editor allows you to enter only the printable ASCII character set (hexadecimal codes 20 to 7E). To force the generation of "control" codes (hexadecimal 00 to 1F and 7F) the ESCAPE character must be used. When you type the ESCAPE key a special character (a reverse video symbol) is placed on the screen. This is treated like any any other character except that the following character becomes a control character. This means that only the lower five bits of code are written to the file thereby generating a code in the range 00 to 1F. An exception is the following: to generate a code of 7F you must enter ESC ?; this sets the seventh bit.</p> <p>To insert the Escape ASCII character (ESC, hexadecimal 1B), type ESC [</p>

Examples

The following table shows some examples of how text can be modified using the functions discussed above. It assumes a text file called EXAMPLE1 on the B: drive.

STEP	IF you enter...	The screen displays...
	EDIT B: EXAMPLE1	— The purpose of this text is to act as an example of how to use the editing functions of the Video File Editor
1	CURSOR DOWN DELETE LINE	as an example of how to use the editing functions of the Video File Editor
2	CURSOR UP ENTER	— as an example of how to use the editing functions of the Video File Editor
3	This is SPACE	This is — as an example of how to use the editing functions of the Video File Editor
4	JOIN LINES	This is as an example of how to use the editing functions of the Video File Editor
5	DELETE CHAR DELETE CHAR DELETE CHAR	This is an example of how to use the editing functions of the Video File Editor

VIDEO FILE EDITOR

STEP	IF you enter...	The screen displays...
6	NEXT LINE	This is an example of how to use <u>the</u> editing functions of the Video File Editor
7	DELETE LINE	This is an example of how to use <u>the</u> Video File Editor
8	RESTORE LINES NEXT LINE	This is an example of how to use the Video File Editor <u>the</u> editing functions of
9	T	This is an example of how to use the Video File Editor <u>The</u> editing function of
10	END OF LINE	This is an example of how to use the Video File Editor The editing functions of__
11	BACKSPACE BACKSPACE	This is an example of how to use the Video File Editor The editing functions __
12	RECALL LINE	This is an example of how to use the Video File Editor the editing functions <u>of</u>
13	SPLIT LINE	This is an example of how to use the Video File Editor the editing functions __ of

STEP	IF you enter...	The screen displays...
14	CURSOR UP	This is an example of how to use the Video File Editor — the editing functions of
15	INSERT LINE	This is an example of how to use the Video File Editor — the editing functions of

Note: To delete a character in the 80th column you should move the cursor to that position in overstrike mode and type **SPACE**.

WINDOW MOVING FUNCTION KEYS

The function keys described in the following table enable you to move the window up and down the file.

FUNCTION KEY	MEANING
HOME (TOP)	Moves the window to the top of the text file. The cursor is placed on the top bar of the file.
END (BOTTOM)	Moves the window to the end of the file. The cursor is placed on the last line of text.

VIDEO FILE EDITOR

FUNCTION KEY	MEANING
PG UP (FULL SCREEN UP)	Causes the window to be moved up the file by 20 lines. This allows one line of overlap between the old and new displays. The cursor remains on the same screen line.
PG DN (FULL SCREEN DOWN)	Causes the window to be moved 20 lines down the file. This allows one line of overlap between the old and new displays. The cursor remains on the same screen line.
SHIFT F10 (HALF SCREEN UP)	Causes the window to be moved half a screen (10 lines) up the file. The cursor remains on the same screen line.
F10 (HALF SCREEN DOWN)	Causes the window to be moved half a screen (10 lines) down the file. The cursor remains on the same screen line.
SHIFT F9 (LINE UP)	Causes the window to be moved one line up the file. The cursor remains on the same screen line.
F9 (LINE DOWN)	Causes the window to be moved one line down the file. The cursor remains on the same screen line.
F6 (NEXT LINE)	Moves the window one line down the file and places the cursor at the start of the next text line.
F7 (GO TO MARK)	Moves the window up or down the file such that the cursor lies on the MARK line.

EXITING AND SAVING FUNCTION KEYS

The function keys described in the following table enable you to exit from the Video File Editor and/or save the file you have been working on.

FUNCTION KEY	MEANING
SHIFT F5 (EXIT AND SAVE)	Causes the revised text to be written back to the file and the Video File Editor to be terminated. The screen is erased and control is returned to MS-DOS.
F5 (SAVE TEXT)	Causes the revised text to be written to the file. The Video File Editor does not terminate.
SHIFT F1 (ABORT)	Causes the Video File Editor to terminate without writing the revised text to the file. If text has been altered or added since starting the editor you are asked to "Confirm Abort?". To confirm press Y. Any other action causes the Video File Editor to ignore the ABORT. Control is returned to MS-DOS.

COMMANDS AND SEARCHING

The second line of the screen (above the scale line) is called the editor command line and is used for entering high level commands and search strings.

To enter text on the editor command line you must first press the **COMMAND MODE** function key. This moves the cursor to the second line. You can now enter text there. All line editing operations such as **INSERT MODE**, **BACKSPACE** and **DELETE CHAR** - now apply to the editor command line. The **RECALL LINE** function when used in command mode restores the editor command line to its previous contents. The **ENTER** key performs **EXECUTE COMMAND** when used in this mode.

Repeating the **COMMAND MODE** key returns the cursor to the next window without performing any command operation.

STRING SEARCHES

This feature enables you to search the file for a particular combination or characters. Before searching for a text you must enter command mode by striking the **COMMAND MODE** function key. Then enter the text to be searched for followed by the appropriate function key, as described in the following table:

FUNCTION KEY	MEANING
F8 (SEARCH DOWN)	Searches for the text string starting from the the current cursor position and moving down the file until the first occurrence of the string is encountered. If found, the window and cursor are moved to it.
SHIFT F8 (SEARCH UP)	Searches for the text string starting from the cursor position and moving up the file. If the string is found then the window and cursor are moved to it.

Examples

The following table shows some examples of the use of the searching functions. It assumes a text file called EXAMPLE2 on the B: drive. When you enter at the A> prompt: EDIT B:EXAMPLE2

If you enter on the editor command line...	Then strike function key...	The screen displays...
	F1	This is an example of how to use the search function keys of the Video File Editor to find a particular combina- tion of characters
func	SEARCH DOWN	This is an example of how to use the search function keys of the Video File Editor to find a particular combina- tion of characters
e SPACEof	SEARCH UP	This is an example of how to use the search function keys of the Video File Editor to find a particular combina- tion of characters

COMMANDS

The Video File Editor commands are a set of special commands that enable you to perform a number of high level functions. Before entering a command you must press the COMMAND MODE function key (F1) to move the cursor to the command line. You can then enter the command which is subsequently displayed on the editor command line. To execute the command you must then press the ENTER key. If you decide not to execute the command press F1 again to return the cursor to the edit text.

GOTO

This command enables you to move the window to a specific line number in the file.

GOTO *line*

Where

SYNTAX ELEMENT	MEANING
<i>line</i>	A decimal integer that is the desired line number in the file. If this number is greater than the number of lines in the file then the window is moved to the end of the file.

Characteristics

Each line of the text file is automatically numbered. That is, the first line of the file is line 1, the TOP bar is line 0 and the MARK bar does not count.



DELETE

This command removes all text between the current line and the MARK line and places the removed text in the restore buffer from where it can be re-inserted at will. If the MARK line does not exist an error message is given.

DELETE



FILE

The FILE command allows you to suspend processing of the current file and invoke the editor on another file. When editing of the new file is terminated by a SAVE AND EXIT or ABORT function, the old file is recalled at the point at which it was exited.

FILE *pathname*

Where

SYNTAX ELEMENT	MEANING
<i>pathname</i>	The path of the new file to be edited.

Characteristics

The command line option flags (/B, /T or /R) used by the old file remain the same for the new file.

Editing of each file is kept entirely independent except for the restore buffer, which enables the transfer of lines of text from one file to another.

Further files can be entered and edited from the new file using the FILE command. There is no limit to the number of levels that can be created in this way except that the text of all the files invoked must fit into the available memory.

7. LINE EDITOR (EDLIN)

ABOUT THIS CHAPTER

This chapter tells you how to use the Line Editor EDLIN.

CONTENTS

INTRODUCTION	7-1	W (WRITE LINES)	7-35
HOW TO INVOKE THE EDLIN PROGRAM	7-1	INTRA-LINE COMMANDS	7-36
EDLIN	7-2	COPY1	7-37
		COPYTO	7-38
INTER-LINE COMMANDS	7-5	COPYLINE	7-40
line (EDIT LINE)	7-6	SKIP1	7-41
A (APPEND LINES)	7-8	SKIPTO	7-42
C (COPY LINES)	7-9	KILL	7-44
D (DELETE LINES)	7-12	INS	7-46
E (END EDITING)	7-15	NEWTEMP	7-47
I (INSERT LINES)	7-16		
L (LIST TEXT)	7-20		
M (MOVE LINES)	7-23		
P (PAGE)	7-25		
Q (QUIT EDITING)	7-26		
R (REPLACE TEXT)	7-27		
S (SEARCH TEXT)	7-30		
T (TRANSFER LINES)	7-34		

INTRODUCTION

The Line Editor (EDLIN) can be used to edit files that contain lines of text, where each line is a maximum of 255 characters, the last of which must be **ENTER**. The files can contain ASCII text or source programs.

Within EDLIN, two types of commands may be used:

- Commands that enable you to perform editing operations on specified lines, a range of lines or an entire file in order to:
 - list, edit, delete and insert lines of text
 - search for a specified text string
 - search for and replace a specified text string
 - create, edit and save new files
 - edit an existing file, save the modified file and keep a back-up of the original file

These commands are termed "inter-line" commands.

- Commands that enable you to perform editing operations within a line of text. These are termed "intra-line" commands and utilize the source line facility as described in Chapter 2.

The control keys described in Chapter 2 can also be used within EDLIN.

In the disk files, the lines of text are not numbered. But when a file is displayed, lines are numbered dynamically. When you create or edit a file, line numbers begin at 1 and are incremented by one through to the end of the file. If you insert new lines between existing lines, all line numbers following the inserted text are automatically incremented by the number of lines inserted. When lines are deleted, all line numbers following the deleted text are decremented automatically by the number of lines deleted. Consequently, lines are always numbered consecutively, starting from 1, through to the last line in the file.

HOW TO INVOKE THE EDLIN PROGRAM

EDLIN

The line editor (EDLIN) is invoked as follows:

[d:][path] EDLIN filespec

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where EDLIN is to be found.
<i>path</i>	Specifies the directory where EDLIN is to be found.
<i>filespec</i>	The file specifier of the file to be edited. This must include the drive specifier unless you want to default to the disk containing the EDLIN command, in which case you need to remove any write-protection.

LINE EDITOR (EDLIN)

Characteristics

When you invoke EDLIN, the back-up file to the file you specify (if it exists) is erased to ensure there is sufficient room on disk for the output file.

If the file specified exists and is smaller than 75% of the available memory, then the entire file is loaded into memory and the following is displayed:

End of input file

*
_

You can then edit the file. Note that the EDLIN prompt is an asterisk (*), and the cursor is an underline (_).

If the file specified exists and is larger than 75% of the available memory, then only the first part of the file is loaded, until 75% of the available memory is full. The EDLIN prompt (*) and cursor (_) will then appear but not the "End of input file" message. You can then edit that part of the file loaded into memory. To access unloaded lines you must use the Write Lines and Append Lines commands described later in this chapter.

If the specified file does not exist on the drive then a new file is created with the specified name. But note that the drive you wish the output file to be written to must be specified when you invoke EDLIN, otherwise the output file will be written to the default drive. The following message is displayed:

New file

*
_

You can then begin to create the file.

There are two edit commands that can be used to terminate the edit session:

- End Edit, which terminates EDLIN, renames the input file filename.BAK and writes the edited file in memory to the output file which is given the same name as the input file. See the "E (END EDIT)" command later in this chapter.

- Quit Edit, which terminates EDLIN without creating a back-up or an output file. The input file remains unchanged. See the "Q (QUIT EDIT)" command later in this chapter.

Note that a file with the extension .BAK cannot be edited. Any attempt to do so will generate the message:

Cannot edit .BAK file--rename file

You must rename the file using the RENAME command (See Chapter 5), then invoke EDLIN on the renamed file.

If, when attempting to create a new file, the following message appears:

No room in directory for file

then either:

- the file directory is full, or
- you have specified an illegal disk drive or file name.

The latter can be checked by examining the command line. (If the command line is no longer on the screen it can be recalled using the F3 (COPYLINE) edit key.) To check the former you can run the CHKDSK command on the specified disk drive. See Chapter 5 for details.

INTER-LINE COMMANDS

This section describes the EDLIN commands that operate on entire lines of text.

The lines you wish to work on may be specified either by entering a line number as a parameter to the command, or by entering a period (.). The latter indicates that the "current line" is to be worked on.

The current line is the location of the last change to the file. It is not necessarily the last line displayed. The current line is indicated by an asterisk between the line number and the first text character. For example:

```
15:* This is the current line
```

Each command description summarizes the purpose of the command, defines the command syntax and explains each syntax element. This is followed, for each command, by a detailed account of the command characteristics and some working examples.

Remarks

1. Commands can be entered in either upper or lower case
2. Command keywords and command parameters can be separated from each other by spaces or commas for readability but need not be, except where two line numbers are entered as parameters, in which case they must be separated by a comma or space. For brevity the syntax of this chapter will always indicate comma where separation is obligatory, but note that a space can alternatively be used
3. Commands only become effective after entering **ENTER**
4. If you make a syntax error when entering a command the message "Entry Error" will be displayed. You must re-enter the command using the correct syntax

line (EDIT LINE)

Enables you to edit a specified line.

[*line* | .]

Where

SYNTAX ELEMENT	MEANING
<i>line</i>	The number of the line to be edited.
.	The current line is to be edited.

Characteristics

When you enter a line number followed by **ENTER** EDLIN displays the line number and the corresponding text, then, on the next screen line, reprints the line number followed by the EDLIN prompt (*) and the cursor (___) . The displayed line serves as the source line and is ready to be edited.

If, instead of a number, you enter a period (.), the current line is displayed and ready for editing. If you enter **ENTER** without a line number or a period, then the line immediately following the current line is displayed, unless the current line is the last one in the file, in which case the edit prompt (*) will re-appear.

LINE EDITOR (EDLIN)

To edit the line you can use any of the control or edit (intraline) keys, or re-enter the entire line by entering text.

If you wish to abort the operation without changing the source line, press **CTRL C** . Pressing **ENTER** with the cursor at the start of the line also aborts the operation without changing anything.

To save the edited line and overwrite the original line, type with the cursor at the end of the edited line. If you type with the cursor in any position other than the start or end of the line, text to the left of the cursor will be written to the file in memory, but all text under and to the right of the cursor will be lost.

Example

Assuming that the contents of the current edit file are as follows:

- 1: This file demonstrates how
- 2: the line command can
- 3: be used to edit line
- 4: four.

IF you type...	THEN EDLIN displays...
4 ENTER	4: *four. 4: * _
INS number SPACE INS F3 ENTER	4: *four. 4: *number four.__ *

A (APPEND LINES)

Adds lines from the input file on disk to that part of the file currently in memory.

[*n*] A

Where

SYNTAX ELEMENT	MEANING
<i>n</i>	The number of lines to be added to the file in memory from the input file on disk.

Characteristics

This command is only useful for files that are too large to fit into the available memory.

When EDLIN is invoked on a file that is too large to fit into memory, it loads as many lines as possible (as much as will fit into 75% of the available memory). Before using the Append command it is therefore necessary to write some lines of text to the output file on disk by means of the Write Lines command (see later).

If you enter A without a parameter, lines are appended from the disk file until the available memory is 75% full, or until there are no more lines to append.

LINE EDITOR (EDLIN)

Any attempt to read beyond the end of the disk file will yield the following message:

End of input file

Example

IF you enter...	THEN...
100 A	the next 100 lines of the input file on disk are read into memory and appended to that part of the file that is already there.

C (COPY LINES)

Copies a range of lines to a specified line.

[line-a] , [line-b] , line-c[, count]C

Where

SYNTAX ELEMENT	MEANING
<i>line-a</i>	The first line in the range to be copied.

SYNTAX ELEMENT	MEANING
<i>line-b</i>	The last line in the range to be copied.
<i>line-c</i>	The line at which the copied lines are to start.
<i>count</i>	The number of times the range is to be copied.

Characteristics

If *line-a* is omitted, then the first line defaults to the current line.

If *line-b* is omitted, then the last line defaults to the current line.

If both *line-a* and *line-b* are omitted, then only the current line is copied.

Following the copy operation, lines that previously followed *line-c* are moved to follow the copied block.

If *line-c* is beyond the current end of file then the lines are copied to line numbers contiguous to the end of file.

line-b must be greater than or equal to *line-a*.

On completion *line-c* becomes the current line.

Examples

Assuming the contents of the current edit file are as follows:

- 1: *This is a sample file
- 2: to demonstrate the use
- 3: of the Copy lines command.

LINE EDITOR (EDLIN)

IF you enter...	THEN the edited file becomes...
1,3,4C	<ul style="list-style-type: none">1: This is a sample file2: to demonstrate the use3: of the Copy lines command.4:*This is a sample file5: to demonstrate the use6: of the Copy lines command
„27,3C	<ul style="list-style-type: none">1: This is a sample file2: to demonstrate the use3: of the Copy lines command.4: This is a sample file5: to demonstrate the use6: of the Copy lines command.7:*This is a sample file8: This is a sample file9: This is a sample file
2,3,8C	<ul style="list-style-type: none">1: This is a sample file2: to demonstrate the use3: of the Copy lines command.4: This is a sample file5: to demonstrate the use6: of the Copy lines command.7: This is a sample file8:*to demonstrate the use9: of the Copy lines command.10: This is a sample file11: This is a sample file

D (DELETE LINES)

Deletes all lines within a specified range.

[*line-a*][,*line-b*] **D**

Where

SYNTAX ELEMENT	MEANING
<i>line-a</i>	The first line in the range to be deleted.
<i>line-b</i>	The last line in the range to be deleted.

Characteristics

If *line-a* is omitted, then the first line defaults to the current line.

If *line-b* is omitted, then the specified line only is deleted.

If **D** is entered alone, then only the current line is deleted.

After the command has been executed the numbers of the lines following the deleted section are changed to follow the numbers of the lines preceding the deleted section.

LINE EDITOR (EDLIN)

The line immediately following the deleted section becomes the current line.

line-b must be greater than (or equal to) *line-a* .

Example

Assuming the contents of the current edit file are as follows:

- 1: This is a sample file
- 2: to demonstrate the use
- 3: of the Delete command.
- 4: "line-a D" deletes just the specified line
- 5: if ",line-b D" is specified then
- 6: all lines from the current line
- 7: to line-b, inclusive, are deleted.
- 8: Specifying both line-a and
- 9: line-b causes that range
- 10: to be deleted.
- 11: Specifying D alone deletes the current line.
- 12: The line subsequent to the
- 13: deleted line(s) becomes the
- 14: current line.

IF you enter...	THEN the edited file becomes...
8,10 D	<ol style="list-style-type: none">1: This is a sample file2: to demonstrate the use3: of the Delete command.4: "line-a D" deletes just the specified line5: if ",line-b D" is specified then6: all lines from the current line7: to line-b, inclusive, are deleted.8:*Specifying D alone deletes the current line.9: The line subsequent to the10: deleted line(s) becomes the11: current line.

IF you enter...	THEN the edited file becomes...
4 D	1: This is a sample file 2: to demonstrate the use 3: of the Delete command. 4:*if ",line-b D" is specified then 5: all lines from the current line 6: to line-b, inclusive, are deleted. 7: Specifying D alone deletes the current line. 8: The line subsequent to the 9: deleted line(s) becomes the 10: current line.
,6 D	1: This is a sample file 2: to demonstrates the use 3: of the Delete command. 4:*Specifying D alone deletes the current line. 5: The line subsequent to the 6: deleted line(s) becomes the 7: current line.
D	1: This is a sample file 2: to demonstrate the use 3: of the Delete command. 4:*The line subsequent to the 5: deleted line(s) becomes the 6: current line.

E (END EDITING)

Exits EDLIN and saves the edited file on disk.

E

Characteristics

The edited file is written to the drive selected when EDLIN was invoked, or to the default drive if no drive was specified. The input file is renamed "filename.BAK". If the file was created during the editing session, no back-up file is created.

You must make sure that enough free space is available on disk to take the output file, otherwise only a portion (at most) will be saved. The remainder will be lost and the message:

Disk full--write not completed

will be displayed, and EDLIN will exit.

Example

IF you enter...	THEN...
E	The current edit file is saved on the diskette specified when EDLIN was invoked. The input file is renamed "filename.BAK", and EDLIN is exited.

I (INSERT LINES)

Allows you to insert lines of text before the specified line number.

[*line* | . | #] I

Where

SYNTAX ELEMENT	MEANING
<i>line</i>	The number of the line before which subsequently entered text is to be inserted.
.	Subsequent text is to be inserted before the current line.
#	Subsequent text is to be appended to the file.

Characteristics

If a line number is entered, then subsequently entered text is inserted immediately before the specified line. If, instead of a line number you enter a period (.), or you enter I on its own, then subsequent text is inserted before the current line. Moreover, if you enter # instead of the line number, text will be appended to the file.

LINE EDITOR (EDLIN)

EDLIN remains in insert mode until either **CTRL Z** or **CTRL C** is entered. While you are in insert mode, successive line numbers appear each time **ENTER** is pressed.

When you exit insert mode, the line immediately following the last inserted line becomes the current line. The line numbers of this and all subsequent lines are automatically incremented by the number of lines inserted.

When a file is created, you must enter **I** before entering text. In this case the line number is 1.

Example

Assuming that the contents of the current edit file are as follows:

```
1: This is a sample file
2: to demonstrate
3: using the Insert Lines
4: command
```

IF you enter..	THEN...
3 I	EDLIN enters insert mode and displays: 3:*__
how lines of text	EDLIN displays: 3:*how lines of text
can be inserted into a file	4:*can be inserted into a file
at a specified line number	5:*at a specified line number 6:*__

IF you enter...	THEN...
CTRL Z	EDLIN exits insert mode.
L	<p>The List command (see later) is invoked and EDLIN displays:</p> <p>1: This is a sample file 2: to demonstrate 3: how lines of text 4: can be inserted into a file 5: at a specified line number 6:*using the Insert Lines 7: command</p>
I or .I	<p>EDLIN enters insert mode and displays:</p> <p>6:* __</p>
or before the current line	<p>EDLIN displays:</p> <p>6: or before the current line 7: * __</p>
CTRL Z	EDLIN exits insert mode.
L	<p>The List command is invoked and EDLIN displays:</p> <p>1: This is a sample file 2: to demonstrate 3: how lines of text 4: can be inserted into a file 5: at a specified line number 6: or before the current line 7:*using the Insert Lines 8: command</p>

LINE EDITOR (EDLIN)

IF you enter...	THEN...
# I or 9	EDLIN enters insert mode and displays: 9: * __
and how lines of text can be appended to a file.	EDLIN displays: 9: and how lines of 10: text can be appended 11: to a file. 12: *
CTRL Z	EDLIN exits insert mode.
L	The List command is invoked and EDLIN displays: 1: This is a sample file 2: to demonstrate 3: how lines of text 4: can be inserted into a file 5: at a specified line number 6: or before the current line 7: using the Insert Lines 8: command 9: and how lines of 10: text can be appended 11: to a file.

L (LIST TEXT)

Displays a specified range of lines.

[line-a] *[, line-b]* L

Where

SYNTAX ELEMENT	MEANING
<i>line-a</i>	The first line in the range to be listed.
<i>line-b</i>	The last line in the range to be listed.

Characteristics

If you specify both *line-a* and *line-b*, then the entire range of lines is displayed, unless this is in excess of 23 lines, in which case the display starts from *line-a*, but this and subsequent lines are scrolled off the top of the screen until *line-b* appears on the 23rd line.

If *line-a* is omitted, but *line-b* is specified, then the display starts 11 lines before the current line and ends at *line-b*. If this is more than 23 lines, the screen scrolls down the file until *line-b* appears on the 23rd line.

LINE EDITOR (EDLIN)

If *line-b* is omitted, but *line-a* is specified, then 23 lines are displayed, starting from *line-a*.

If you enter L on its own, then the current line appears on the center line of the screen (unless the current line is less than line 12), with the preceding 11 lines displayed before it, and the subsequent 11 lines displayed after it. If the current line is before line 12, EDLIN displays the first 23 lines.

Example

Assuming the contents of the current edit file are as follows:

```
1: This is a sample file
2: to demonstrate the
3: use of the List
4: command
.
.
.
14:*This is the current line
.
.
.
23: The List command can be
24: used to examine
25: different parts of the
26: file, up to 23 lines
27: at once.
```

IF you enter...	THEN EDLIN displays...
2,4 L	2: to demonstrate the 3: use of the List 4: command

IF you enter...	THEN EDLIN displays...
,24 L	3: use of the List 4: command . . . 14: *This is the current line . . . 23: The List command can be 24: used to examine
24 L	24: used to examine 25: different parts of the 26: file, up to 23 lines 27: at once.
L	3: use of the List 4: command . . . 14: *This is the current line 24: used to examine 25: different parts of the

M (MOVE LINES)

Moves a range of lines to a specified line.

[line-a] , *[line-b]* , *[line-c]* M

Where

SYNTAX ELEMENT	MEANING
<i>line-a</i>	The first of the range of lines to be moved.
<i>line-b</i>	The last of the range of lines to be moved.
<i>line-c</i>	The line to which the text is to be moved.

Characteristics

If *line-a* is omitted, then the first line defaults to the current line.

If *line-b* is omitted, then the last line defaults to the current line.

line-b must be greater than or equal to *line-a*.

If *line-a* is omitted, *line-b* can be specified as the relative number of lines forward of the current line, by preceding the number with a + .

Following the move, lines are renumbered depending on the direction of the move. For instance, moving lines 10 to 20 to line 100 would effectively delete lines 10 to 20 (thereby causing all subsequent lines to be moved up the file 11 lines), then the moved lines would become lines 79 to 99.

The first of the moved lines becomes the current line.

Examples

Assuming the contents of the current edit file are as follows:

- 1: This is a sample file
- 2: to demonstrate the use
- 3: of the Move lines command.
- 4: New first line

100: Next line.

IF you enter...	THEN the edit file becomes...
1,3,100 M	1: New first line . . . 97: This is a sample file 98: to demonstrate the use 99: of the Move lines command. 100: Next line.

Lists a specified range of lines.

[line-a][, line-b] P

Where

SYNTAX ELEMENT	MEANING
<i>line-a</i>	The first of the range of lines to be displayed.
<i>line-b</i>	The last of the range of lines to be displayed.

Characteristics

If *line-a* is omitted, then the first line defaults to the line following the current line.

If *line-b* is omitted 23 lines are displayed.

The last line displayed becomes the current line.

The difference between the Page command and the List command is that the Page command changes the current line.

Q (QUIT EDITING)

Quits the editing session but does not save any changes you have made.

Q

Characteristics

After entering the Quit command EDLIN replies with the message:

Abort edit (Y/N)?__

Pressing **N** or any key other than **Y** or **CTRL C** continues the editing session. Pressing **Y** terminates the editing session. No BAK file is created, and any changes made during the editing session are lost. The file on disk remains exactly as it was when EDLIN was invoked. Note also that any previous BAK file is also lost since the current BAK file is always deleted when EDLIN is invoked.

Example

IF you enter...	THEN...
Q	the editing session is terminated without saving the changes made during the editing session.

R (REPLACE TEXT)

Searches a specified range to replace all occurrences of one string with another string.

```
[line-a][,line-b] [ ? ] R string-a[ CTRL Z string-b]
```

Where

SYNTAX ELEMENT	MEANING
<i>line-a</i>	The number of the first line in the range on which the Replace Text command is to be executed.
<i>line-b</i>	The number of the last line in the range.
?	For each occurrence of the specified string the O.K.? prompt appears, enabling you to accept or reject the replacement.
<i>string-a</i>	The string of characters that are to be replaced.
<i>string-b</i>	The string of characters that are to replace <i>string-a</i> .

Characteristics

For each line in which a replacement occurs the modified line is displayed on the screen. If you entered the ? parameter then the prompt:

O.K.?

will appear after each replacement. You must then enter **Y** or **ENTER** to confirm the replacement, or strike any other key to reject it. In either case the search will recommence for the next occurrence of *string-a*. If ? is not specified, all occurrences of *string-a* will be replaced by *string-b* without confirmation.

If you omit *string-b*, then all occurrences of *string-a* are deleted.

If you omit *line-a* then the search will begin from line 1. If you omit *line-b* then the search will continue to the end of the file in memory. If neither *line-a* nor *line-b* is entered, then the entire file in memory will be searched and modified.

Once all replacements have been made, the Replace Text command terminates and the last line in which *string-a* occurred becomes the current line.

If the replacement string causes a line to expand beyond the limit of 254 characters then the following message is displayed:

Line too long

Example

Assuming that the contents of the current edit file are as follows:

- 1: This is a sample file
- 2: to demonstrate the
- 3: use of the Replace Text
- 4: command.
- 5: Using this command a
- 6: specified group of characters
- 7: can be replaced by
- 8: another group of characters
- 9: *and can be deleted entirely.

LINE EDITOR (EDLIN)

IF you enter...	THEN...
5,8 R group of characters CTRL Z string	<p>all occurrences of "group of characters" in the range line 5 to line 8 are replaced with "string". EDLIN displays:</p> <p>6: specified string 8: another string</p> <p>and line 8 becomes the current line</p>
? Rand CTRL Z or	<p>each occurrence of "and" within the entire file can optionally be replaced with "or". EDLIN displays:</p> <p>4: commor. O.K.?</p>
N	<p>5: using this commor a O.K.?</p>
N	<p>9: or can be deleted entirely? O.K.?</p>
Y	<p>* _ and line 9 becomes the current line</p>
L	<p>the List command is executed to display the file and enable you to see the changes you have made. EDLIN displays:</p> <p>1: This is a sample file 2: to demonstrate the 3: use of the Replace Text 4: command. 5: Using this command a 6: specified string 7: can be replaced by 8: another string 9: *or can be deleted entirely.</p>

S (SEARCH TEXT)

Searches a specified range of lines for a specified string.

[line-a][, line-b] [?] S string

Where

SYNTAX ELEMENT	MEANING
<i>line-a</i>	The number of the line from which the search is to start.
<i>line-b</i>	The number of the last possible line to be searched.
<i>?</i>	On finding a matching string the: O.K.? prompt is to be displayed, thereby enabling you to accept or reject the particular occurrence.
<i>string</i>	The string of characters to be searched for.

Characteristics

The command searches the range *line-a* to *line-b* for the specified string. If *line-a* is not specified then the search begins from the line after the current line. If *line-b* is not specified then the last line in the range is the last line of the file in memory.

When a matching string is found, the corresponding line is displayed. Then if the ? parameter is not specified, the search is terminated and the displayed line becomes the current line. If ? is specified, then the displayed line will be followed by the prompt:

O.K.?

To accept the string you must enter either **Y** or **ENTER**. The search will then terminate and the displayed line becomes the current line. Striking any other key, however, will re-commence the search for the next occurrence of the string.

If no matching string is found, or if ? is specified and all matching strings are rejected, the message:

Not found

is displayed.

Example

Assuming that the contents of the current edit file are as follows:

- 1: This is a sample file to demonstrate
- 2: The use of the Search Text command.
- 3: The search can either display the
- 4: first occurrence of a specified string
- 5: and terminate, or, if specified
- 6: to do so, it will enable you to
- 7: interactively examine each
- 8: occurrence of a string allowing
- 9: you to confirm or reject the string.
- 10: Once a string is accepted the
- 11: *search terminates.

IF you enter...	THEN...
2,5 Sstring	<p>the Search Text command will search lines 2 to 5 of the file, inclusive, for the first occurrence of the string "string", then display the following:</p> <p>4: first occurrence of a specified string</p> <p>and terminate. Line 4 becomes the current line.</p>
,8 Sstring	<p>the search will be made on lines 5 (one after the current line) to 8, inclusive, for the first occurrence of "string". EDLIN will display:</p> <p>8: occurrence of a string allowing</p> <p>and the search will terminate with line 8 as the current line.</p>
1? Sstring N Y	<p>the Search Text command searches the file for the first occurrence of "string". The search starts from line 1. The result is:</p> <p>4: first occurrence of a specified string O.K.?</p> <p>The string is rejected and the search continues for the next occurrence. The result is:</p> <p>8: occurrences of a string allowing O.K.?</p> <p>The search is terminated and line 8 becomes the current line.</p>

LINE EDITOR (EDLIN)

IF you enter...	THEN...
Ssample file	<p>the Search Text command searches for the string "sample file" starting from line 9 (one after the current line), up to the end of the file. The string is not found, hence the message:</p> <p>Not found</p> <p>is displayed. The search terminates and line 8 remains the current line.</p>

T (TRANSFER LINES)

Inserts an entire file before a specified line of the current edit file.

[line] T *[filespec]*

Where

SYNTAX ELEMENT	MEANING
<i>line</i>	The number of the line before which the file is to be inserted.
<i>filespec</i>	The file to be inserted.

Characteristics

If the line parameter is omitted then the current line is assumed.

The specified file must be in the same directory as the edit file.

W (WRITE LINES)

Writes a specified number of lines from the file being edited in memory to the output file on disk.

[*n*] W

Where

SYNTAX ELEMENT	MEANING
<i>n</i>	The number of lines to be written to diskette, starting from line 1. If this parameter is omitted, then lines of text are written to the output file until the available memory is 25% full.

Characteristics

The Write Lines command is used in conjunction with the Append Lines command when editing files that are too large to fit into the available memory. Lines written to the output file are deleted from memory, and the remaining lines renumbered, starting from line 1. This leaves space available at the end of the file in memory for additional lines to be read from the input file on disk using the Append Lines command.

INTRA-LINE COMMANDS

The intra-line commands are executed using the special editing keys that can be used to perform edits within the current line taking advantage of the source line facility. They enable you to:

- copy one character from the source line to the current line (COPY1)
- copy a specified portion of the source line to the current line (COPYTO)
- copy all remaining characters in the source line to the current line (COPYLINE)
- delete a specified character in the source line (SKIP1)
- delete a specified portion of the source line (SKIPTO)
- kill the current input and delete the source line (KILL)
- enter insert mode to insert text into the current line (INS)
- exit insert mode (enter overstrike mode) (INS)
- make the current line the source line (NEWTEMP)

Moreover, you can also use the MS-DOS control keys when in EDLIN.

For details about control keys refer to Chapter 2.

The remainder of this section describes each of the intra-line commands in turn.

Before starting to edit the current line, a copy of the current line exists in the source line. You then begin editing the current line by entering an edit line by entering text and by using the intra-line commands. Not until you complete the edit line by pressing **ENTER** does the edit line replace the contents of the current line.

You can select a line to work on using the line command described in the previous section.

Copies one character from the source line to the edit line.

F1

Characteristics

Pressing the **F1** key copies one character from the source line to the edit line. Insert mode, if active, is automatically turned off.

Example

Assuming that the line to be edited is displayed as follows:

```
1:*This is the COPY1 command
1:*__
```

IF you enter...	THEN...
F1	the first character is copied from the source line into the edit line thus: 1:*This is the COPY1 command 1:*T__
F1	the next character is copied from the source line into the edit line thus: 1:*This is the COPY1 command 1:*Th__



COPYTO

Copies up to a given character from the source line to the edit line.

F2 *character*

Where

SYNTAX ELEMENT	MEANING
<i>character</i>	A character whose first occurrence in the source line will terminate the copy operation. If the character does not appear in the source line nothing will be copied.

Characteristics

Pressing the **F2** key copies all characters up to but not including a given character from the source line to the edit line. The cursor is moved to the position of the given character. The given character is not displayed.

LINE EDITOR (EDLIN)

Example

Assuming that the line to be edited is displayed as follows:

1:*This is the COPYTO command

1:*__

IF you enter...	THEN...
F2 c	the characters "This is the COPYTO " are copied from the source line to the edit line thus: 1:*This is the COPYTO command 1:*This is the COPYTO __

COPYLINE

Copies the source line to the edit line.

F3

Characteristics

Pressing the **F3** key copies all remaining characters from the source line to the edit line regardless of cursor position. Following the copy, the cursor is positioned after the last character on the line. Insert mode, if active, is automatically turned off.

Examples

Assuming that the line to be edited is displayed as follows:

1:*This is the COPYLINE command

1:* _

IF you enter...	THEN...
F3	all remaining characters are copied from the source to the edit line thus: 1:*This is the COPYLINE command 1:*This is the COPYLINE command__

Skip one character in the source line.

DEL

Characteristics

Pressing the **DEL** key skips over one character in the source line without copying it to the edit line. It does not affect insert mode.

Example

Assuming that the line to be edited is displayed as follows:

```
1:*This is the SKIP1 command
1:*
```

IF you enter...	THEN...
DEL	you skip the letter "T" in the source line.
F3	the remaining characters are copied from the source line to the edit line thus: 1:*This is the SKIP1 command 1:*his is the SKIP1 command__



SKIPTO

Skip to specified character in the source line.

F4 *character*

Where

SYNTAX ELEMENT	MEANING
<i>character</i>	The character in the source line that terminates the string that is to be skipped.

Characteristics

Pressing the **F4** key causes characters in the source line to be skipped from the edit character up to but not including the first occurrence of the given character. If the source line does not contain the given character then no characters are skipped. Nothing is copied to the edit line by this command. Insert mode remains unaffected.

LINE EDITOR (EDLIN)

Examples

Assuming that the line to be edited is displayed as follows:

1:*This is the SKIPTO command

1:*__

IF you enter...	THEN...
F4 c	all characters in the source line up to the first "c" are skipped over.
F3	the remaining characters in the source line are copied to the edit line thus: 1:*This is the SKIPTO command 1:*command__



KILL

Clear the edit line.

ESC

Characteristics

Pressing the **ESC** key clears the edit line, but the source line remains unchanged. The KILL command also displays a back-slash (\) and inserts a carriage return and a line-feed. The cursor is placed immediately under the first character of the terminated line. You can then begin again to edit the line. Insert mode is turned off by this command.

Examples

Assuming your current and edit lines are displayed as follows:

1:*This is the KILL command

1:*This is the__

LINE EDITOR (EDLIN)

IF you enter...	THEN...
ESC	the edit line is emptied thus: 1:*This is the KILL command 1:*This is the\ —
F3	the source line is copied to the edit line thus: 1:*This is the KILL command 1:*This is the\ This is the KILL command

INS

Enters/exits insert mode.

INS

Characteristics

Pressing the **INS** key enters or exits insert mode. On entering insert mode, subsequently entered characters will be inserted before the character under the cursor when insert mode was entered.

Examples

Assuming the line to be edited is displayed as follows:

```
1:*This is INS command
1:*This is __
```

IF you enter...	THEN...
INS the	the characters "the" are inserted in the edit line thus: 1:*This is INS command 1:*This is the __
INS F3	insert mode is switched off and the remainder of the source line is copied to the edit line thus: 1:*This is INS command 1:*This is the INS command__

Creates a new source line by copying the edit line to the source line.

F5

Characteristics

Pressing the **F5** key copies the edit line to the source line. The original contents of the source line are deleted. An "@" sign appears at the end of the edit line and a carriage return line-feed is inserted. The edit line is also cleared and insert mode is exited.

Example

Assuming the source and edit lines are displayed as follows:

1:*This is the NEWLINE command

1:*__

and you want to change the source line to read "This is the NEWTEMP command":

IF you enter...	THEN...
F2 L	<p>all characters up to the first "L" are copied from the source line to the edit line thus:</p> <p>1:*This is the NEWLINE command 1:*This is the NEW__</p>
TEMP	<p>the next four characters in the source line are replaced in the edit line by the characters "TEMP" thus:</p> <p>1:*This is the NEWLINE command 1:*This is the NEWTEMP__</p>
F3	<p>the remaining characters in the source line are copied to the edit line thus:</p> <p>1:*This is the NEWLINE command 1:*This is the NEWTEMP command__</p>
F5 F3	<p>the contents of the source line are replaced with those of the edit line. The display appears thus:</p> <p>1:*This is the NEWLINE command 1:*This is the NEWTEMP command@ This is the NEWTEMP command</p>

8. THE LINKER

ABOUT THIS CHAPTER

This chapter describes the LINK utility.

CONTENTS

INTRODUCTION	8-1
TEMPORARY FILES	8-1
CHANGING DISKETTES	8-2
SEGMENTS, GROUPS AND CLASSES	8-2
INTERACTIVE ENTRY	8-3
COMMAND LINE ENTRY	8-10
AUTOMATIC RESPONSE FILE ENTRY	8-13

INTRODUCTION

MS-LINK is an executable program available on your MS-DOS system diskette. It links separate object modules that are the output of the assembler or a compatible compiler, resolving external references by searching multiple library files. Its output is a relocatable run file, along with a list file that shows external references and error messages.

To run MS-LINK you must provide appropriate object, run, list and library file parameters. In addition, you may enter switches that modify the way in which MS-LINK processes your input. Parameters and switches are fully described in the section "Interactive Entry".

There are three related methods of running MS-LINK. These are described later in this chapter in the following sections:

- Interactive Entry
- Command Line Entry
- Automatic Response File Entry

Interactive entry is the primary method and its section contains all the information common to the three methods.

TEMPORARY FILES

MS-LINK uses available memory for the link session. If the files to be linked create an output file that exceeds available memory, MS-LINK creates a temporary file on the default drive and names it VM.TMP. If MS -LINK needs to create VM.TMP, it displays the message:

**VM.TMP has been created.
Do not change diskette in drive x:**

Once this message is displayed, you must not remove the diskette from the default drive until the link session ends. If the diskette is removed, the operation of MS-LINK is unpredictable, and MS-LINK might return the error message:

Unexpected end of file on VM.TMP

MS-LINK uses VM.TMP as virtual memory. The contents of VM.TMP are subsequently written to the file name following the run file prompt. VM.TMP is a working file only and is deleted at the end of the linking session.

If the default drive already has a file by the name of VM.TMP, it will be deleted by MS-LINK and a new file will be allocated; the contents of the previous file are destroyed. You should therefore avoid using VM.TMP as one of your own file names.

CHANGING DISKETTES

You may wish to change diskettes during the link operation. For example, if MS-LINK cannot find an object file on the specified diskette it prompts you to change diskettes instead of aborting the session. Or if you enter the /PAUSE switch, MS-LINK pauses and prompts you to change diskettes before creating the run file. You may change diskettes when prompted except in the following cases:

- When the diskette you wish to change has a VM.TMP file created on it (see the previous section)
- When you have requested a list file on the diskette you wish to change

SEGMENTS, GROUPS AND CLASSES

Some of the terms used in this chapter are explained below to help you understand how MS-LINK works. Generally, if you are linking object modules compiled from BASIC, Pascal, or any high-level language, you will not need to know these terms. If you are writing and compiling programs in assembly language, however, you will need to understand MS-LINK and the definitions described below.

In MS-DOS, memory can be divided into segments, classes, and groups. For example:

Group Contents	Segment Names	Segment Class Names
Segment 1 Segment 2 Segment 12	PROG.1 PROG.2 PROG.3	CODE CODE DATA

THE LINKER

Note that segments 1, 2, and 12 have different segment names but may or may not have the same segment class name. Segments 1, 2, and 12 form a group with a group address of the lowest address of segment 1 (that is, the lowest address in memory).

Each segment has a segment name and a class name. MS-LINK loads all segments into memory by class name from the first class encountered to the last. All segments assigned to the same class are loaded into memory contiguously.

During processing, MS-LINK references segments by their addresses in memory. MS-LINK does this by finding groups of segments.

A group is a collection of segments that fit within a 64K byte area of memory. The segments do not need to be contiguous to form a group. The address of any group is the lowest address of the segments in that group. At link time, MS-LINK analyzes the groups, then references the segments by the address in memory of that group. A program may consist of one or more groups.

If you are writing in assembly language, you may assign the group and class names in your program. In high-level languages (BASIC, COBOL, FORTRAN, Pascal), the naming is done automatically by the compiler.

INTERACTIVE ENTRY

With interactive entry MS-LINK prompts you for each parameter in turn.

Type the following:

`[d:][path] LINK`

Four prompts appear, one at a time, requesting the appropriate parameters. These are summarized in the following table.

PROMPT	RESPONSE
Object Modules [OBJ]:	<i>objfile</i> [+ <i>objfile</i>] ... [<i>switch</i>] ...
Run File [objfile.EXE]:	[<i>runfile</i>] [<i>switch</i>] ...
List File [NUL.MAP]:	[<i>listfile</i>] [<i>switch</i>] ...
Libraries [.LIB]:	[<i>libfile</i> [+ <i>libfile</i>] ...] [<i>switch</i>] ...

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where LINK is to be found.
<i>path</i>	Specifies the directory where LINK is to be found.
<i>objfile</i>	The file specification of an object module to be linked. Such specifications must be separated from each other with a plus sign (+) or a SPACE . The default file extension is .OBJ. If any extension is different from .OBJ it must be specified; otherwise it may be omitted. Segments are loaded by class name, from the first class encountered to the last. The order in which you list object files is therefore significant.

SYNTAX ELEMENT	MEANING
<i>runfile</i>	The file specification of the run (executable) file that results from the link session. All run files receive the extension .EXE. Any other extension you may enter is ignored. The default run file name is the first object file name entered.
<i>listfile</i>	The file specification of the list file that contains an entry for each segment in the object files and its offset in the run file. The default list file name is the NUL file. The default extension is .MAP.
<i>libfile</i>	The file specification of each library. Up to eight libraries may be searched; separate each specification with a plus sign (+) or a SPACE . The default is standard library search. The default extension is .LIB. Libraries are searched in the order they are listed.
<i>switch</i>	The name of any of the switches described in the table below. You may enter any number of switches, at the end of any number of prompt responses (before pressing ENTER). Switches may be abbreviated to the / together with the first letter or any sequential substring starting with the first letter.

Note

That [d:][path] may precede any file name mentioned.

SWITCH	MEANING
/DSALLOCATE (or /D)	All data defined to be in DGROUP is loaded at the high end of the group. If the switch is not used MS-LINK loads all data at the low end of the group. At runtime the data space pointer is set to the lowest possible address, allowing the entire storage to be used. Use of the /DSALLOCATE switch in combination with the default load low (that is, the /HIGH switch is not used) permits the user application to allocate dynamically any available memory below the area specifically allocated within DGROUP, yet to remain addressable by the same data space pointer. This dynamic allocation is needed for Pascal and FORTRAN programs. The maximum amount of memory that can be allocated by the application is 64K (or the amount actually available) minus the allocated portion of DGROUP.
/HIGH (or /H)	MS-LINK loads the run file as high as possible in memory. If /HIGH is not specified the run file is loaded as low as possible. DO NOT USE THIS SWITCH WITH PASCAL OR FORTRAN PROGRAMS.
/LINENUMBERS (or /L)	MS-LINK includes in the list file the line numbers and addresses of the source statements in the input modules. If /LINENUMBERS is not specified, line numbers and addresses are not included. (Not all compilers produce object modules that contain line number information. In these cases, of course MS-LINK cannot include line numbers.)

SWITCH	MEANING
/MAP (or /M)	<p>MS-LINK lists all public (global) symbols defined in the output modules. If /MAP is not given, MSLINK will list only errors (including undefined globals).</p> <p>The symbols are listed alphabetically. For each symbol, MS-LINK lists its value and its segment: offset location in the run file. The symbols are listed at the end of the list file.</p>
/NODEFAULTLIBRARY SEARCH (or /N)	<p>MS-LINK does not automatically search the default library to resolve external references. For example, linking Pascal object modules with the /N switch stops MS-LINK automatically searching the file PASCAL.LIB.</p>
/PAUSE (or /P)	<p>MS-LINK pauses in the link session when the switch is encountered. Normally, MS-LINK performs the linking session without stopping from beginning to end. /PAUSE allows the user to swap diskettes before MS-LINK outputs the run (.EXE) file.</p> <p>When MS-LINK encounters the /PAUSE switch, it displays the message:</p> <p style="text-align: center;">About to generate .EXE file Change disks - hit any key</p> <p>MS-LINK resumes processing when the user presses any key. DO NOT SWAP A DISKETTE THAT IS TO RECEIVE A LIST FILE, OR A DISK USED FOR A TEMPORARY (VM.TMP) FILE.</p>

SWITCH	MEANING
/STACK:size (or /S)	<p>The size of the stack provided for the load module by the assembler or compiler is overridden. The stack size becomes that specified in the 'size' parameter, which must follow the switch name and a colon.</p> <p>If a value from 1 to 511 is entered, MS-LINK uses 512. At least one object (input) module must contain a stack allocation statement. If not, MS-LINK will return a WARNING: NO STACK SEGMENT error message.</p>

Characteristics

After any of these responses, before pressing **ENTER** , you may enter a comma (,) followed by the answer to what would be the next prompt, without having to wait for that prompt.

If you conclude any response with a semicolon (;) the remaining responses are all assumed to be the default. Linking begins immediately with no further prompting.

Use the plus sign (+) not only to separate lists of object files and libraries but to extend these lists, where necessary, onto more than one line. Enter the plus sign followed by **ENTER** at the end of a physical line. This repeats the object file or library prompt, and enables you to continue the logical line with further file names.

Example

This sample shows you a typical dialog for an MS-LINK session.

In response to the MS-DOS prompt, enter:

LINK

THE LINKER

The system displays the following messages, prompts and your responses:

Microsoft 8086 Object Linker
(C) Copyright Microsoft Corp.

Object Modules [.OBJ]: IO SYSINIT
Run File [IO.EXE]:
List File [NUL.MAP]: IO /MAP
Libraries [.LIB]:

Fig. 8-1 LINK Example

Notes:

1. By specifying /MAP, you get both an alphabetic listing and a chronological listing of public symbols.
2. By responding PRN to the List File: prompt, you can redirect your output to the printer.
3. By specifying the /LINE switch, MS-LINK gives you a listing of all line numbers for all modules. (Note that the /LINE switch can generate a large volume of output).
4. By pressing **ENTER** in response to the Libraries: prompt, an automatic library search is performed.

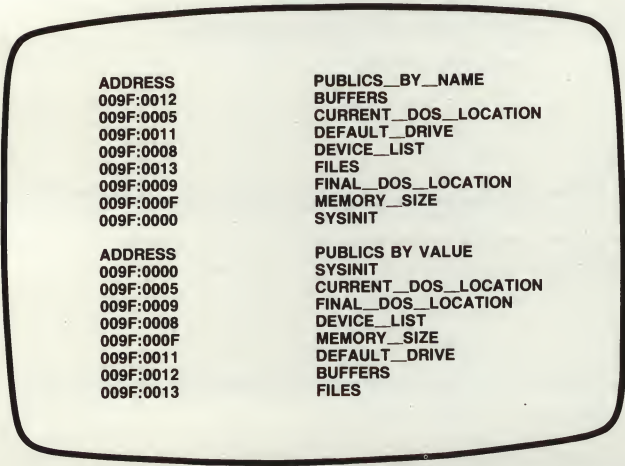
Once MS-LINK locates all libraries, the linker map displays a list of segments in the order of their appearance within the load module. The list might look like this:

Start	Stop	Length	Name
00000H	009ECH	09EDH	CODE
009FOH	01166H	0777H	SYSINITSEG

The information in the Start and Stop columns shows the 20-bit hex address of each segment relative to location zero. Location zero is the beginning of the load module.

The addresses displayed are not the absolute addresses where these segments are loaded.

Because the /MAP switch was used, MS-LINK displays the public symbols by name and value. For example:



ADDRESS	PUBLICS_BY_NAME
009F:0012	BUFFERS
009F:0005	CURRENT_DOS_LOCATION
009F:0011	DEFAULT_DRIVE
009F:0008	DEVICE_LIST
009F:0013	FILES
009F:0009	FINAL_DOS_LOCATION
009F:000F	MEMORY_SIZE
009F:0000	SYSINIT

ADDRESS	PUBLICS BY VALUE
009F:0000	SYSINIT
009F:0005	CURRENT_DOS_LOCATION
009F:0009	FINAL_DOS_LOCATION
009F:0008	DEVICE_LIST
009F:000F	MEMORY_SIZE
009F:0011	DEFAULT_DRIVE
009F:0012	BUFFERS
009F:0013	FILES

Fig. 8-2 /MAP Sample Display

COMMAND LINE ENTRY

With command line entry you enter the MS-LINK command along with its parameters, without waiting to be prompted.

You must separate each complete parameter entry from the next with a comma (,). Apart from this, what you enter is the same as with interactive entry. The command syntax is therefore the following:

[d:][path] **LINK** *objfile* [+ *objfile*] ... [*switch*] ... [, [*runfile*] [*switch*] ... [, [*listfile*] [*switch*] ... [, [*libfile*] [+ *libfile*] ...] [*switch*] ...]]]

Where

SYNTAX ELEMENT	MEANING
<i>d:</i>	Specifies the drive where LINK is to be found.
<i>path</i>	Specifies the directory where LINK is to be found.
<i>objfile</i>	An object module to be linked.
<i>runfile</i>	The run (executable) file to be created.
<i>listfile</i>	The list file to be output.
<i>libfile</i>	A library file to be searched.
<i>switch</i>	A switch to be applied.

Note

[d:][path] may precede any filename.

See the section "Interactive Entry" for full descriptions of these parameters.

Characteristics

Use the plus sign (+) as with interactive entry not only to separate lists of object files and library files but to extend those lists, where necessary, onto more than one line.

You may enter switches after any one of the four parameter entries (that is, before any of the commas or the final **ENTER**).

To accept the default parameter for a syntax element, enter a second comma with no space between the two commas. To accept a further default enter a third comma and so on. Remember that you must make an entry for the object file parameter.

If you enter a semicolon (;) at any time, the unspecified parameters all assume default values. Linking begins immediately.

If you enter an incomplete list of parameters and no semicolon is used, MS-LINK prompts you for the next remaining entry. See "Interactive Entry" for a full list of MS-LINK prompts.

Examples

IF you enter...	THEN...
LINK FUN + TEXT + TABLE + CARE/P/M,,FUNLIST,COBLIB.LIB	MS-LINK is loaded, then the object modules FUN.OBJ, TEXT.OBJ, TABLE.OBJ and CARE.OBJ are loaded. MS-LINK then pauses (because of the /P switch). The object modules are linked when you press any key, and a global symbol map is produced (because of the /M switch). The default run file FUN.EXE is created, along with a list file called FUNLIST. MAP. The library file COBLIB. LIB is searched for external references.
LINK FUN,,	MS-LINK is loaded, then the object module FUN.OBJ is loaded. The default run file FUN.EXE is created. MS-LINK then prompts for a list file, offering the file FUN.MAP as default.

AUTOMATIC RESPONSE FILE ENTRY

With automatic response file entry you enter the name of a file that already contains the answers to some or all of the MS-LINK parameter prompts. Precede this file name by the symbol @. The command syntax is therefore the following:

d:*[path]* **LINK** @[*drive*:] [*f-path*] *filespec*

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where LINK is to be found.
<i>path</i>	Specifies the directory where LINK is to be found.
<i>filespec</i>	The file specification of the file that contains responses to the MS-LINK prompts. Use of a file extension is optional; there is no default extension.
<i>drive</i>	Specifies the drive where <i>filespec</i> is to be found.
<i>f-path</i>	Specifies the directory where <i>filespec</i> is to be found.

Characteristics

Automatic response files can contain several lines of text, each corresponding to an MS-LINK prompt. Responses must be in the same order as with interactive input.

Press **ENTER** to indicate the conclusion of each response and the beginning of the next. Type a plus sign (+) followed by **ENTER** to continue a response to the object module or libraries prompt on a new line. See the section "Interactive Entry" for full details of each MS-LINK prompt.

Use switches, commas and colons in an automatic response file just as you would in interactive entry.

You can enter the name of more than one automatic response file on the command line, and combine response file names with additional parameters. The combined series of resulting parameters must be a valid sequence of MS-LINK prompts.

When the MS-LINK session begins, each prompt is displayed in order with the responses from the response file. If the response file does not contain answers for all the prompts, (in the form of file names, the semicolon command character or carriage returns), MS-LINK displays the prompt which does not have a response, then waits for you to enter a legal response. When a legal response has been entered, MS-LINK continues the link session.

Example

IF you enter...	THEN...
<pre>COPY CON C:\COB\INC\INC1 FUN + TEXT + TABLE + CARE /PAUSE/MAP FUNLIST COBLIB.LIB F6 C:\COB\LINK@C:\INC\INC1</pre>	<p>this response file tells MS-LINK to load the four object modules named FUN, TEXT, TABLE, and CARE. MS-LINK pauses before producing a public symbol map to permit you to swap diskettes. When you press any key, the output files are named FUN.EXE and FUNLIST.MAP. MS-LINK searches the library file COBLIB.LIB.</p>

9. THE DEBUGGER

ABOUT THIS CHAPTER

This chapter describes the DEBUG utility.

CONTENTS

INTRODUCTION	9-1	P (PROCEED)	9-32
HOW TO INVOKE THE DEBUG PROGRAM	9-1	Q (QUIT)	9-33
DEBUG	9-1	R (REGISTER)	9-34
DEBUGGING COMMANDS	9-4	S (SEARCH)	9-37
COMMAND PARAMETERS	9-4	T (TRACE)	9-39
A (ASSEMBLE)	9-7	U (UNASSEMBLE)	9-40
C (COMPARE)	9-10	W (WRITE)	9-42
D (DUMP)	9-11		
E (ENTER)	9-14		
F (FILL)	9-17		
G (GO)	9-18		
H (HEX)	9-21		
I (INPUT)	9-23		
L (LOAD)	9-24		
M (MOVE)	9-27		
N (NAME)	9-29		
O (OUTPUT)	9-31		

INTRODUCTION

The DEBUG utility is a program that provides a controlled testing environment for binary and executable object files. It eliminates the need to re-assemble a program to see if a problem has been corrected by a minor change. Moreover, it enables you to change the contents of a file or CPU register, then to immediately re-execute a program to check the validity of the changes.

HOW TO INVOKE THE DEBUG PROGRAM

**DEBUG**

The DEBUG program is invoked as follows:

[d:][path] DEBUG [(drive:)] [file-path] filespec [, arglist]

Where

SYNTAX ELEMENT	MEANING
<i>d</i>	Specifies the drive where DEBUG is to be found.
<i>path</i>	Specifies the directory where DEBUG is to be found.
<i>drive</i>	Specifies the drive where <i>filespec</i> is to be found.
<i>file-path</i>	Specifies the directory where <i>filespec</i> is to be found.
<i>filespec</i>	The specifier of the program file to be debugged.

SYNTAX ELEMENT	MEANING
<i>arglist</i>	<p>A list of file name parameters and switches separated by commas. These will be passed to the program specified by the <i>filespec</i> parameter. Thus, when the program is loaded into memory, it is loaded as if it had been invoked with the command:</p> <p style="text-align: center;"><i>filespec arglist</i></p> <p>That is, <i>filespec</i> indicates the file to be debugged, and <i>arglist</i> is the rest of the command line that is used when the file is invoked and loaded into memory.</p>

Characteristics

On entering the DEBUG environment DEBUG responds with the hyphen (-) prompt and underline (__) cursor. You are then free to enter any DEBUG command.

If you enter DEBUG without parameters, since no file name has been specified, current memory, disk blocks, or disk files can be worked on using other debugging commands.

If you include the *filespec* in the command line then the specified file is loaded into memory starting at location 100 (hexadecimal). However, if the file has an EXE extension, then it is relocated to the address specified in the header of the file. Moreover, if the file has the HEX extension, then the file is loaded beginning at the address specified in the HEX file.

THE DEBUGGER

Examples

IF you enter...	THEN...
DEBUG	the DEBUG environment is entered, but without loading a file.
DEBUG b:myprog	the DEBUG environment is entered and the file named "myprog" is loaded into memory from drive B.
DEBUG progs\dev\myrtn, prog1,prog2	the DEBUG environment is entered and the file named "myrtn" is loaded into memory from the sub-directory "dev" of directory "progs". The loaded file also takes two file name parameters: "prog1" and "prog2".

Remarks

When you invoke DEBUG, it sets up a program header at offset 0 in the program work area. You can overwrite this area if you enter DEBUG without parameters. Moreover, if you are debugging a file with a COM or EXE extension you must not tamper with the program header below location 5CH, or DEBUG will terminate.

Do not restart a program after the "Program terminated normally" message is displayed. You must reload the program with the N and L commands for it to run properly.

DEBUGGING COMMANDS

This section describes the DEBUG commands in alphabetical order for ease of reference. Each such command description summarizes the purpose of the command, defines the command syntax and explains each syntax element. This is followed, for each command, by a detailed account of the command characteristics and some working examples.

Remarks

1. Commands can be entered in either upper or lower case.
2. Command keywords and command parameters can be separated from each other by spaces or commas for readability but need not be, except where two hexadecimal numbers are entered as parameters, in which case they must be separated by a comma or space. For brevity, the syntax of this chapter will always indicate comma where separation is obligatory, but note that a space can alternatively be used.
3. Commands only become effective after pressing **ENTER** .
4. If you make a syntax error when entering a command the message "Error" will be displayed. You must re-enter the command using the correct syntax.

COMMAND PARAMETERS

The following DEBUG command parameters require special definition.

THE DEBUGGER

PARAMETER	DEFINITION
<i>address</i>	<p>A hexadecimal value in one of the following formats:</p> <ul style="list-style-type: none">• A segment register designation and a hexadecimal offset separated from each other by a colon. For example: DS:0300• A hexadecimal segment and offset separated from each other by a colon. For example: 9D0:0100• A hexadecimal offset value. For example: 200 <p>The DEBUG command will append a default segment value from either the DS or CS registers, depending on the command.</p>
<i>byte</i>	A one or two character hexadecimal value.
<i>drive</i>	0, 1 or 2 depending on whether you wish to select drive A, drive B or drive C, respectively.

PARAMETER	DEFINITION
<i>range</i>	<p>A range of addresses specified as:</p> <p>EITHER:</p> <p style="padding-left: 40px;">address L value</p> <p>where address specifies the start of the range and value specifies the length of the range. For example:</p> <p style="padding-left: 40px;">DS:300L30</p> <p>indicates a range of 48 locations starting at address 300 in the sector indicated by the DS register.</p> <p>The specified range cannot be greater than 1000 (hexadecimal). To specify this value enter 0000 (or 0) as the value parameter.</p> <p>OR:</p> <p style="padding-left: 40px;">address,address</p> <p>where the two addresses indicate the limits of the range. Note that space may be used instead of comma.</p>
<i>value</i>	<p>A 1 to 4 character hexadecimal value.</p>

A (ASSEMBLE)

Assembles assembler mnemonics directly into memory.

A [*address*]

Where

SYNTAX ELEMENT	MEANING
<i>address</i>	The start address at which the subsequently entered line of mnemonics is to be assembled. If this parameter is omitted, location 100 is assumed.

Characteristics

After you enter the Assemble command, DEBUG displays the specified *address* followed by the cursor. You may then enter a line of assembler mnemonics. On terminating the line with **ENTER** the line will be assembled into memory starting at the specified location. The address of the byte subsequent to the assembled code will be displayed on the next line along with the cursor to enable you enter the next line of code. If, instead of a line of assembler mnemonics, you simply press **ENTER**, the Assemble command terminates and the DEBUG prompt re-appears.

All numeric values are hexadecimal and must be entered as 1 to 4 characters. Prefix mnemonics must be specified in front of the opcode to which they refer. You may also enter them on a separate line.

The segment override mnemonics are CS:, DS:, ES: and SS:. The mnemonic for the far return is RETF. String manipulation mnemonics must explicitly state the string size. For example, use MOVSW to move word strings and MOVSB to move byte strings.

The Assemble command will automatically assemble short, near or far jumps and calls, depending on byte displacement with respect to the destination address. These may be overridden with the NEAR or FAR prefix. For example:

```
0100:0500 JMP 502           ;a two-byte short jump
0100:0502 JMP NEAR 505      ;a three-byte near jump
0100:505  JMP FAR 50A       ;a five-byte far jump
```

The NEAR prefix may be abbreviated to NE, but the FAR prefix cannot be abbreviated.

DEBUG cannot tell whether some operands refer to a word memory location or to a byte memory location. In this case the data type must be explicitly stated with the prefix "WORD PTR" or "BYTE PTR". Acceptable abbreviations are "WO" and "BY". For example:

```
NEG BYTE PTR [128]
DEC WO [SI]
```

A simple operand is a literal. Whereas operands enclosed within square brackets refer to memory. For example:

```
MOV AX,21           ;Load AX with 21H
MOV AX,[21]         ;Load AX with the contents of location
                   ;21H
```

Two popular pseudo-instructions are available with the Assemble command. The DB opcode will assemble word values directly into memory. For example:

```
DB 1,2,3,4,"THIS IS AN EXAMPLE"
DB 'THIS IS A QUOTE: "'
DB "THIS IS A QUOTE'"

DW 1000,2000,3000,"BACH"
```


THE DEBUGGER

The Assemble command supports all forms of register indirect commands. For example:

```
ADD BX,34[BP + 2].[SI-1]
POP [BP + DI]
PUSH [SI]
```

All opcode synonyms are also supported. For example:

```
LOOPZ 100
LOOPE 100
JA 200
JNBE 200
```

Example

IF you enter...	THEN...
A200	DEBUG displays 09AC:0200 __
MOV AX,[21]	the assembler mnemonics are assembled starting at location 200. The byte location subsequent to the assembled code is then displayed thus 09AC:0203 __
ENTER	the Assemble command terminates and the DEBUG prompt re-appears.

C (COMPARE)

Compares the contents of two areas of memory.

C *range* , *address*

Where

SYNTAX ELEMENT	MEANING
<i>range</i>	The range of addresses defining the first area to be compared. If no segment is specified then the segment specified in the DS register is assumed.
<i>address</i>	The start of the area to be compared with the area specified by the range parameter.

Characteristics

The Compare command compares the area of memory specified by the *range* parameter with an area of the same size starting at the location specified by the *address* parameter.

If the contents of the two areas are identical nothing is displayed. If there are differences, then the differences are displayed in the form:

address1 contents1 contents2 address2

THE DEBUGGER

where: address1 indicates the address in the first area and contents1 its contents; and address2 indicates the corresponding address in the second area and contents2 its contents.

Example

IF you enter...	THEN...
C100, 1FF, 300 or C100L100, 300	the area of memory from 100 to 1FF is compared with the area of memory from 300 to 3FF.

D (DUMP)

Displays an area of memory.

D [*range* | *address*]

Where

SYNTAX ELEMENT	MEANING
<i>range</i>	The range of addresses whose contents are to be displayed. If you enter only an offset, then the segment specified in the DS register is assumed.
<i>address</i>	The address from which the display is to start. The contents of this address and the subsequent 127 locations are displayed.

Characteristics

If D is specified without parameters then the 128 bytes following the last address to be displayed are displayed. If no location has yet been accessed then the dump will start from location DS:100.

If D and the *range* parameter are specified then the contents of that range of addresses are displayed. If this takes more than 24 screen lines the display is scrolled until the contents of the final address in the range are displayed on line 24.

If D and a single address are specified, then the contents of the 128 locations starting from the specified address are displayed.

The dump is displayed in two portions:

- A hexadecimal dump, where each byte is represented by its hexadecimal value.
- An ASCII dump, where the equivalent ASCII character for the byte is displayed. If there is no corresponding printable ASCII character then the dump displays a period (.).

Each line of the dump begins with an address which is followed by the hexadecimal contents of the 16 bytes starting from the addressed location. The eighth and ninth bytes are separated by a hyphen (-). The right-hand columns display the equivalent ASCII values. Each line of the display, except possibly the first, begins on a 16 byte boundary.

Examples

IF you enter...	THEN...
D 100,110 or D100L11	a hexadecimal and ASCII dump of lines 100 to to 110 (hexadecimal), inclusive, are displayed.
D	a hexadecimal and ASCII dump of the 128 bytes starting from location 111 (hexadecimal) is displayed.
D200	a hexadecimal and ASCII dump of the 128 bytes starting from location 200 (hexadecimal) is displayed.

E (ENTER)

Replaces the contents of memory locations at the byte addresses specified.

E *address* [, *bytevalue* [, *bytevalue*]...]

Where

SYNTAX MEANING	MEANING
<i>address</i>	The address of the location whose value is to be replaced; or the address of the first of a succession of locations whose contents are to be replaced. If only an offset is specified then the segment indicated by the DS register is assumed.
<i>bytevalue</i>	The hexadecimal byte value that is to replace the contents of the specified address. The first <i>bytevalue</i> parameter will replace the contents of the location specified by the address parameter. A second <i>bytevalue</i> will replace the contents of the location following that specified by the address parameter, and so on.

Characteristics

If the command is entered without the *bytevalue* list, then DEBUG displays the specified *address* and its contents. The Enter command then waits for you to perform one of the following:

- Replace the displayed byte value by entering another value. You simply enter the new value after the current value. If you enter an illegal value, or if you type more than two digits then the illegal or extra character is not echoed.
- Advance to the next byte by pressing **SPACE** . To change the value of this byte simply enter the value as described above. If you advance beyond an eight-byte boundary, DEBUG starts a new display line with the address displayed at the start of the line. To advance to the next byte without changing the current byte simply press **SPACE** again.
- Return to the previous byte by entering hyphen (-). On doing so DEBUG starts a new display line with the address of the byte you have returned to and its contents. You can then change the contents of this location as described above. To move back one byte further without changing this value simply enter hyphen again, and another new display line will be generated.
- Terminate the Enter command by pressing **ENTER** . This key may be pressed in any byte position.

If you specify *bytevalues* in the command line then the first of these *bytevalues* will replace the contents of the location specified by the *address* parameter. Subsequent entries in the list of *bytevalues* will replace subsequent bytes in memory.

Examples

IF you enter...	THEN...
E100	DEBUG displays something like: 058D:0100 CD.____
26 SPACE	the value of location 100 is changed to 26 and DEBUG displays: 058D:0100 CD.26____ the next byte (location 101) is displayed: 058D:0100 CD.26 20.____
SPACE	the next byte (location 102) is displayed: 058D:100 CD.26 20. 00.____
-	the previous byte (location 101) is displayed on the next line: 058D:0100 CD.26 20. 00. 058D:0101 20.____
30	the contents of location 101 are changed to 30 and the Enter command is terminated: 058D:0100 CD.26. 20. 00. 058D:0101 20.30 >__
E200,26,0A,19,23	the contents of byte locations 200, 201, 202 and 203 are changed to 26, 0A, 19 and 23, respectively

Fills an area of memory with specified byte values.

F *range* , *bytevalue*[, *bytevalue*...]

Where

SYNTAX ELEMENT	MEANING
<i>range</i>	The range of addresses whose contents are to be overwritten with the specified byte values. If only the offset is specified then the segment indicated by the DS register is assumed.
<i>bytevalue</i>	A two digit hexadecimal value that is to overwrite the contents of the specified address.

Characteristics

If the specified *range* contains more bytes than the list of *bytevalues* , then the list of byte values is repeated until the specified range is filled.

If the list of *bytevalues* is longer than the specified *range* then the extra *bytevalues* are ignored.

Example

IF you enter...	THEN...
F04BA:100L100,42,45,48,37,20	DEBUG fills memory locations 04BA:100 to 04BA:1FF with the byte values specified. The five values are repeated until all 256 locations are filled.

G (GO)

Executes the program currently in memory, optionally halting at a specified breakpoint and displaying information about the system/program environment.

G [= *address*][, *address*]...

Where

SYNTAX ELEMENT	MEANING
= <i>address</i>	The address in memory at which program execution is to start. " = " must be entered to distinguish a start address from a breakpoint address.

THE DEBUGGER

SYNTAX ELEMENT	MEANING
<i>address</i>	The breakpoint address at which program execution is to halt and the register and flag states to be displayed along with the next instruction to be executed.

Characteristics

If you enter G without parameters then the program currently in memory is executed starting from the address specified by the CS and IP registers.

If you specify the *= address* parameter, then the contents of the CS and IP registers are changed to those specified by the *= address* parameter, and the program in memory is executed, starting from that point.

If you specify one or more breakpoint addresses then program execution stops at the first such address encountered and displays the contents of the registers, the state of the flags and the next instruction to be executed (see the Register command for a description of the display).

You may specify up to ten breakpoint addresses, in any order. If your program has many paths you can use this feature to ensure that your program halts, whichever path it takes.

If you enter more than ten breakpoints DEBUG will display:

BP Error

Before executing the program the Go command replaces the contents of the breakpoint locations with an interrupt instruction (hexadecimal CC). When program execution halts at such a location DEBUG restores the original values of all the specified breakpoint locations. However, if the program terminates normally (that is, not at a specified breakpoint), then the breakpoint values are not restored.

Each breakpoint address that you specify must point to the first byte of an 8086 instruction, otherwise unpredictable results will occur.

The user stack pointer must have six bytes available for this command, otherwise unpredictable results will occur.

Example

IF you enter...	THEN...
G = 200,1AF,141	the program currently in memory is executed starting from location 200. Assuming location 141 is encountered before 1AF, then the program halts at location 141 and the register and flag values are displayed along with the next instruction to be executed. If neither breakpoint location is encountered, then the program terminates normally.
G	if, in the previous example, the program halted at location 141, then program execution continues from that address. If program execution terminated normally in the previous example, then program execution again starts at location 200.

Calculates and displays the sum and the difference of two hexadecimal values.

H *value-a* , *value-b*

Where

SYNTAX ELEMENT	MEANING
<i>value-a</i>	The first of two hexadecimal values.
<i>value-b</i>	The hexadecimal value that is to be added to or subtracted from <i>value-a</i> .

Characteristics

The hexadecimal values may be up to four characters long.

The Hex command displays two four digit values:

- The first is the result of adding *value-b* to *value-a*
- The second is the result of subtracting *value-b* from *value-a*

Example

IF you enter...	THEN...
H19F,10A	DEBUG displays: 02A9 0095
HFFFF,2	DEBUG displays: 0001 FFFD

I (INPUT)

Inputs and displays (in hexadecimal) one byte from the specified port.

I value

Where


SYNTAX ELEMENT	MEANING
<i>value</i>	The hex address of the port from which the byte is to be input.

Characteristics

The port address can be up to 16 bits.

Example

IF you enter...	THEN...
I2F8	the byte at the addressed port is input and displayed.



L (LOAD)

Loads a file or absolute disk sectors into memory.

L [*address* [, *drive* , *sector-a* , *sectors*]]

Where

SYNTAX ELEMENT	MEANING
<i>address</i>	The address in memory at which the file or specified sectors, is to be loaded. If only an offset is entered then the segment indicated by the CS register is assumed. Sectors cannot be loaded across segments.
<i>drive</i>	The drive from which disk sectors are to be loaded. For drive A you must enter 0, for drive B you must enter 1, or for drive C you must enter 2.
<i>sector-a</i>	The first of a range of sectors to be loaded from the disk specified by the drive parameter.
<i>sectors</i>	The number of sectors to be loaded. The maximum number of sectors that can be specified is 80 Hexadecimal.

THE DEBUGGER

Characteristics

If all parameters are specified then DEBUG loads sectors of information from disk into memory. The first such sector to be loaded is that specified by the *sector-a* parameter. It is loaded at the memory location specified by the *address* parameter. This sector is the first of a continuous range of sectors to be loaded, the number of which is specified by the *sectors* parameter.

If you enter L without parameters, or with just the *address* parameter, the file whose file control block is correctly formatted at location CS:5C is loaded into memory. The file control block at CS:5C is set either to the *filespec* specified when the DEBUG command was invoked, or to the *filespec* specified by the most recent subsequent Name command.

If L is entered alone, then the file is loaded at location CS:100. If you specify L and the *address* parameter, the file is loaded at the specified address. In either case DEBUG sets the BX:CX registers to the number of bytes loaded.

If the file has an .EXE extension, then it is relocated to the load address specified in the loader of the .EXE file. That is, the *address* parameter to the Load command is ignored. Note that the header itself is stripped off the .EXE file before the file is loaded into memory. Thus the size of the .EXE file on disk will differ from its size in memory.

If the file is a .HEX file, then entering the Load command with no parameters causes the file to be loaded starting at the address specified within the .HEX file. If the *address* parameter, however, is specified then loading starts at the address which is the sum of the address specified and the address in the .HEX file.

Examples

The following examples assume the system to be initially in MS-DOS.

IF you enter...	THEN...
debug Nb:file.com L	the debugger is entered and the subsequent Name command sets the file control block at CS:5C to identify file "file.com" on the diskette inserted in drive B. The Load command then loads this file into memory starting at CS:100 (the default address).
debug b:file.com L300	file.com is loaded into memory at location CS: 100 by the DEBUG command. It is then relocated to CS:300 by the Load command.
debug L500,1,OF,6D	109 sectors are loaded into memory from drive B starting from sector OF. They are placed in memory starting at location CS:500.

M (MOVE)

Moves the contents of a specified range of memory addresses to the locations starting at a specified address.

M *range* , *address*

Where

SYNTAX ELEMENT	MEANING
<i>range</i>	The area of memory whose contents are to be moved. If you only entered an offset then the segment indicated in the DS register is assumed.
<i>address</i>	The start of the destination area. If you only entered an offset then the segment indicated by the DS register is assumed.

Characteristics

If the source and destination areas overlap the move is performed without loss of data.

The contents of the source area are not changed by the move, unless the destination area overlaps it.

If you specify an address as the end of the range you must only enter the offset. The segment specified, or defaulted to, in the start address of the range is assumed.

Example

IF you enter...	THEN...
MCS:100,110,CS:500 or MCS:100L11,CS:500	the 11 bytes starting at location CS:100 through to 110 are copied to the 11 bytes starting at location CS:500.

N (NAME)

Provides file names for the Load and Write commands or file name parameters for the program to be debugged.

N [*drive:*][*file-path*] *filespec* [*arglist*]

Where

SYNTAX ELEMENT	MEANING
<i>filespec</i>	The file specifier of a file to be loaded into memory, written to diskette, or used as a file name parameter to the file currently in memory.

See Page 9-1 for the definition of the other syntax elements.

Characteristics

The name command can be used to provide:

- The name of the disk file to be loaded into memory by a subsequent Load command.
- The name to be assigned to the file currently in memory when the file is subsequently written to disk.
- File name parameters to the file in memory to be debugged.

The first case enables you to specify the file you wish to debug after entering the DEBUG environment. That is, you can enter DEBUG without specifying parameters, then use the Name command to name the disk file you wish to debug, then load the file into memory using the Load command. This has the same effect as entering the file name as the first parameter to the DEBUG command. In either case the file control block for the file to be debugged is set up at location CS:5C and the file is loaded.

In the second case the file is already in memory and the Name command sets up the file control block for the specified file name at location CS:5C. When a Write command is subsequently entered the file in memory is written to disk with the file name whose file control block is set up at location CS:5C.

In the third case the Name command provides file name parameters for the program currently in memory. Whatever file control block was set at CS:5C is replaced by that of the first such parameter. If a second file parameter is specified then its file control block is set up at location CS:6C. Only two file control blocks are set up although additional file name parameters may be included if required. All the file names specified are placed in a save area at CS:81, with CS:80 containing a character count. Parameters specified in this way are analogous to file names specified in the argument list to the DEBUG command.

Examples

IF you enter...	THEN...
DEBUG Nb:file.com L	the system enters the DEBUG environment and the file named file.COM resident on drive B has its file control block set up at location CS:5C. The Load command subsequently loads this file into memory. This sequence has the same effect as entering "DEBUG b:file.com".

THE DEBUGGER

IF you enter...	THEN...
Nb:newfile.com W	the file control block is set up at location CS:5C for the file specifier "b:newfile.com". The subsequent Write command writes the file currently in memory to drive B and names the file "newfile.com".
DEBUG b:file1.com Nfile2.dat,file3. dat G	the DEBUG command loads the file named file1.com from drive B to be debugged. The Name command sets up two file control blocks at locations CS:5C and CS:6C for the file specifiers b:file2. dat and b:file3.dat, respectively. These files then become parameters to file1.com when the subsequent Go command executes file1.com.

O(OUTPUT)

Sends a specified byte to an output port.

O *value , byte*

Where

SYNTAX ELEMENT	MEANING
<i>value</i>	The address of the output port. It must be specified in hexadecimal and can be up to 16 bits.
<i>byte</i>	A two-digit hexadecimal value to be sent to the specified port.

Example

IF you enter...	THEN...
01E8, 27	the byte value 27 Hex is output to the port IE8.

P (PROCEED)

Proceed past a CALL or INT instruction.

P [= *address*] [, *value*]

See Page 9-40 for the definition of the syntax elements.

Characteristics

This command's key function is to execute, without tracing, all CALL and INT instructions. In addition it displays repeated instructions only once and traces loops (terminated by LOOP XX) only on the first pass.

Q(QUIT)

Terminates the DEBUG program.

Q

Characteristics

The Quit command terminates the debugger without saving the file you are working on. Control is returned to MS-DOS command mode.

Example

IF you enter...	THEN...
Q	the DEBUG program terminates and returns you to MS-DOS command mode.

R(REGISTER)

Displays the hexadecimal contents of the registers and flag settings, or displays the contents of a specified register with the option to change that value, or displays the flag settings with the option of reversing any number of those settings.

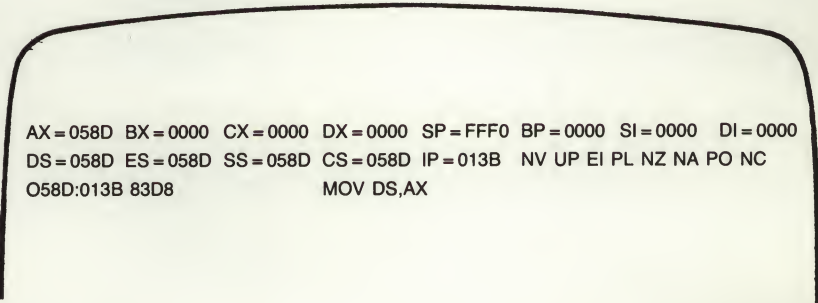
R [*register-name* | F]

Where

SYNTAX ELEMENT	MEANING															
<i>register-name</i>	<p>Any valid register name whose contents are to be examined and optionally changed. This may be one of:</p> <table><tr><td>AX</td><td>DX</td><td>SI</td><td>ES</td><td>IP</td></tr><tr><td>BX</td><td>SP</td><td>DI</td><td>SS</td><td>PC</td></tr><tr><td>CX</td><td>BP</td><td>DS</td><td>CS</td><td></td></tr></table> <p>Note: IP and PC both refer to the Instruction Pointer.</p>	AX	DX	SI	ES	IP	BX	SP	DI	SS	PC	CX	BP	DS	CS	
AX	DX	SI	ES	IP												
BX	SP	DI	SS	PC												
CX	BP	DS	CS													
F	<p>The flag settings are to be displayed and, optionally, changed.</p>															

Characteristics

If you enter R without parameters, then the contents of all registers are displayed along with the flag settings and the next instruction to be executed. For example:



```
AX = 058D BX = 0000 CX = 0000 DX = 0000 SP = FFF0 BP = 0000 SI = 0000 DI = 0000
DS = 058D ES = 058D SS = 058D CS = 058D IP = 013B NV UP EI PL NZ NA PO NC
058D:013B 83D8          MOV DS,AX
```

Fig. 9-1 Sample R Display

If you enter R with a *register-name*, then DEBUG displays the contents of that register. The command then waits for you to do one of the following:

- Press **ENTER** to terminate the Register command without changing the value of the displayed register.
- Change the value of the register by entering the four-digit hexadecimal value then terminate the Register command by pressing **ENTER**.

The valid flag values are shown in the following table:

FLAG NAME	SET	CLEAR
Overflow	OV (yes)	NV (no)
Direction	DN (decrement)	UP (increment)
Interrupt	EI (enabled)	DI (disabled)
Sign	NG (negative)	PL (plus)
Zero	ZR (yes)	NZ (no)
Auxiliary carry	AC (yes)	NA (no)
Parity	PE (even)	PO (odd)
Carry	CY (yes)	NC (no)

If you enter RF, then the current flag settings are displayed. You can then either:

- Press **ENTER** to terminate the Register command without changing the flag values, or
- Change the setting of one or more flags by entering the alternate value of the appropriate flags. The new values may be entered in any order, with or without delimiters.

THE DEBUGGER

Examples

IF you enter...	THEN...
R	DEBUG displays the contents of all registers, flag settings and the next instruction to be executed.
R IP	DEBUG displays the contents of the Instruction Pointer. For example: IP 0139 :___
013B	the contents of the Instruction Pointer are changed to 013B.
RF	DEBUG displays the flag settings. For example: NV UP EI PL NZ NA PO NC-__
PE ZR DI NG	The Parity flag is set to even (PE), the Zero flag is set (ZR), the Interrupt flag is cleared (DI), and the Sign flag is set (NG).
RF	DEBUG displays the new state of the flags: NV UP DI NG ZR NA PE NC-__

S (SEARCH)

Searches a specified range for a list of bytes.

S *range* , *list*

Where

SYNTAX ELEMENT	MEANING
<i>range</i>	The range of addresses within which the search is to be made. If you only enter the offset then the segment indicated by the DS register is assumed.
<i>list</i>	The list of bytes to be searched for. Bytes in the list must be separated by a space or a comma.

Characteristics

For each occurrence of the list of bytes within the specified range, DEBUG returns the address of the first byte. If no address is returned, no match was found.

Example

IF you enter...	THEN...
S100L100,20 or S100,1FF,20	<p>DEBUG displays the address of every occurrence of byte value 20 in the address range 100 to 1FF, inclusive. For example:</p> <pre> 058D: 010C 058D: 0110 058D: 0115 058D: 0118 058D: 0120 058D: 0128 058D: 0125 </pre>

T (TRACE)

Executes one or more instructions and displays the register contents, flag settings and the next instruction to be executed.

T [= *address*][, *value*]

Where

SYNTAX ELEMENT	MEANING
<code>= address</code>	DEBUG is to commence execution at this address.
<code>value</code>	The number of instructions to be executed.

Characteristics

If the `= address` parameter is not specified then execution begins at CS:IP.

If the `value` parameter is not specified then only one instruction is executed.

The display generated is of the same format as that of the Register command (without parameters).

Examples

IF you enter...	THEN...
<code>T = 200,5</code>	five instructions, starting with the one at location 200 are executed, and the register and flag values following each instruction are displayed along with the next instruction to be executed.
<code>T</code>	the instruction pointed to by CS:IP is executed and the register and flag contents are displayed along with the next instruction to be executed.

U (UNASSEMBLE)

Disassembles strings of bytes in memory and displays them as assembler-like statements along with their corresponding addresses.

U [*range*|*address*]

Where

SYNTAX ELEMENT	MEANING
<i>range</i>	The range of addresses whose byte values are to be disassembled. If you do not specify the segment then the segment indicated by the CS register is assumed.
<i>address</i>	The start of a 32 byte area of memory to be disassembled. If you only enter an offset then the segment indicated by the CS register is assumed.

Characteristics

If neither the *range* nor *address* parameter is specified, then 32 bytes are disassembled starting at location CS:IP (16 bytes are disassembled in 40 column screen mode).

The number of bytes disassembled may be slightly more than the number you specified. This is because instructions are not always the same length and the final address in a range will not always contain the last byte of an instruction.

The first address of a range, or the *address* parameter, must always refer to the first byte of an assembler instruction, otherwise results will be unpredictable.

Example

IF you enter...	THEN...
U058D:204L8	eight bytes starting at location 058D:204 are disassembled and the result displayed: 058D:0204 8D16DFOD LEA DX,[0DDF] 058D:0208 42 INC DX 058D:0209 03D0 ADD DX,AX 058D:020B 8916E50B MOV [0BE5],DX

W (WRITE)

Writes the file being debugged to disk.

W [*address* , *drive* , *sector-a* , *sectors*]

THE DEBUGGER

Where

SYNTAX ELEMENT	MEANING
<i>address</i>	The start address of the code in memory that is to be written to disk. If you enter only an offset then the segment indicated in the CS register is assumed.
<i>drive</i>	The drive containing the specified blocks to which code in memory is to be written. For drive A you must enter 0, for drive B you must enter 1, or for drive C you must enter 2.
<i>sector-a</i>	The sector number on disk that is the first of a contiguous range of sectors to be overwritten with code from memory.
<i>sectors</i>	The number of disk sectors blocks to be overwritten with code from memory. The maximum number of sectors that can be specified is 80 Hexadecimal.

Characteristics

If you enter the Write command without parameters, then the file is written to disk starting from memory address CS:100. If you specify the *address* parameter then the file in memory, starting from the specified address, is written to disk.

In either case, before executing the Write command, BX:CX must be set to the number of bytes to be written. This value was set up correctly when the file was loaded (either by the Load command or the DEBUG command itself). However, if, since loading the file, you have executed a Go or Trace command, then the value of BX:CX will have been changed. You must be sure this value is set up correctly.

When the Write command writes a file to disk it obtains the drive specifier and file name via the file control block set up at CS:5C. If no drive specifier is set up then the default is assumed. This file control block is set up either by the DEBUG command or a subsequent Name command. If it does not indicate the file specifier you require, you must set up this file control block using the Name command.

When the file is written to disk it overwrites the version currently on disk unless the specified file name does not exist, in which case a new file is created.

If all parameters are specified then the code in memory is written to the drive specified by the parameter. The data to be written starts at the memory location specified by the *address* parameter, and is written to the blocks on disk specified by the *sector-a* and *sectors* parameters. You must therefore be **extremely** careful to specify the required sectors, since information held there will be destroyed by this operation.

Examples

IF you enter...	THEN...
W	the file in memory, starting from location CS:100, is written to disk with the file specifier defined by the file control block set up at location CS:5C.
W200	the file in memory, starting from location CS:200, is written to disk with the file specifier defined by the file control block set up at location CS:5C.
W200,1,1F,20	20 Hex (32 decimal) sectors on drive B starting from location 1F are overwritten with the data starting at memory location CS:200.

**A. ASCII DISPLAY AND KEYBOARD CODE
TABLES**

ABOUT THIS APPENDIX

This appendix provides a table of ASCII codes and extended keyboard codes.

CONTENTS

BASIC ASCII CODE A-1
(ISO 7-bit code)

EXTENDED ASCII CODE A-5
FOR THE DISPLAY
(ISO 8-bit code)

NATIONAL VARIATIONS A-8
IN EXTENDED ASCII
CODE (ISO 8-bit code)

EXTENDED KEYBOARD A-14

ASCII DISPLAY AND KEYBOARD CODE TABLES

BASIC ASCII CODE (ISO 7-bit code)

This is a 7 bit code used for Information Interchange. The codes 0 to 31 and code 127 (all decimal numbers) are Control Codes. These Control Codes are used for Data Transmission, Device Control and Formatting Effects. Note that CTRL shows on the screen as ^.

DECIMAL	HEXADECIMAL	CHARACTER	CONTROL	MEANING
0	00	(BLANK)	NUL	Null
1	01	CTRL A	SOH	Start Of Heading
2	02	CTRL B	STX	Start of Text
3	03	CTRL C	ETX	End of Text
4	04	CTRL D	EOT	End of Transmission
5	05	CTRL E	ENQ	Enquiry
6	06	CTRL F	ACK	Acknowledge
7	07	CTRL G	BEL	Bell
8	08	CTRL H	BS	Backspace
9	09	CTRL I	HT	Horizontal Tabul.
10	0A	CTRL J	LF	Line Feed
11	0B	CTRL K	VT	Vertical Tabulation
12	0C	CTRL L	FF	Form Feed
13	0D	CTRL M	CR	Carriage Return
14	0E	CTRL N	SO	Shift-Out
15	0F	CTRL O	SI	Shift-In
16	10	CTRL P	DLE	Data Link Escape
17	11	CTRL Q	DC1	Device Control (1)
18	12	CTRL R	DC2	Device Control (2)
19	13	CTRL S	DC3	Device Control (3)
20	14	CTRL T	DC4	Device Control (4)
21	15	CTRL U	NAK	Negative Acknowl.
22	16	CTRL V	SYN	Synchronous Idle
23	17	CTRL W	ETB	End of Trans. Block
24	18	CTRL X	CAN	Cancel
25	19	CTRL Y	EM	End of Medium
26	1A	CTRL Z	SUB	Substitute Charac.
27	1B	CTRL [ESC	Escape
28	1C	CTRL \	FS	File Separator
29	1D	CTRL]	GS	Group Separator
30	1E	CTRL ^	RS	Record Separator
31	1F	CTRL _	US	Unit Separator
32	20	(SPACE)	SP	Space

DECIMAL	HEXADECIMAL	CHARACTER	CONTROL	MEANING
---------	-------------	-----------	---------	---------

33	21	!
34	22	"
35	23	#
36	24	\$
37	25	%
38	26	&
39	27	'
40	28	(
41	29)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G

ASCII DISPLAY AND KEYBOARD CODE TABLES

DECIMAL HEXADECIMAL CHARACTER CONTROL MEANING

72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n

DECIMAL	HEXADECIMAL	CHARACTER	CONTROL	MEANING
---------	-------------	-----------	---------	---------

111	6F	o		
112	70	p		
113	71	q		
114	72	r		
115	73	s		
116	74	t		
117	75	u		
118	76	v		
119	77	w		
120	78	x		
121	79	y		
122	7A	z		
123	7B	{		
124	7C			
125	7D	}		
126	7E			
127	7F	(DELETE)	DEL	Delete previous character

ASCII DISPLAY AND KEYBOARD CODE TABLES

EXTENDED ASCII CODE FOR THE DISPLAY (ISO 8-bit code)

This table shows the 256 elements of the extended ASCII character set, together with their decimal and hexadecimal equivalents.

DEC:	HEX	CHARACTER	DEC:	HEX	CHARACTER	DEC:	HEX	CHARACTER	DEC:	HEX	CHARACTER
000	00	BLANK (NUL)	016	10	► (DLE)	032	20	SPACE (SP)	048	30	0
001	01	☺ (SOH)	017	11	◄ (DC1)	033	21	!	049	31	1
002	02	☹ (STX)	018	12	↕ (DC2)	034	22	"	050	32	2
003	03	♥ (ETX)	019	13	!! (DC3)	035	23	#	051	33	3
004	04	♦ (EOT)	020	14	⏏ (DC4)	036	24	\$	052	34	4
005	05	♣ (ENQ)	021	15	§ (NAK)	037	25	%	053	35	5
006	06	♠ (ACK)	022	16	▬ (SYN)	038	26	&	054	36	6
007	07	• (BEL)	023	17	↕ (ETB)	039	27	'	055	37	7
008	08	■ (BS)	024	18	↑ (CAN)	040	28	(056	38	8
009	09	○ (HT)	025	19	↓ (EM)	041	29)	057	39	9
010	0A	◉ (LF)	026	1A	→ (SUB)	042	2A	*	058	3A	:
011	0B	♂ (VT)	027	1B	← (ESC)	043	2B	+	059	3B	;
012	0C	♀ (FF)	028	1C	└ (FS)	044	2C	,	060	3C	<
013	0D	♪ (CR)	029	1D	↔ (GS)	045	2D	—	061	3D	=
014	0E	♫ (SO)	030	1E	▲ (RS)	046	2E	.	062	3E	>
015	0F	☼ (SI)	031	1F	▼ (US)	047	2F	/	063	3F	?

Tab. A-1 Extended ASCII Character Set

DEC:	HEX	CHARACTER	DEC:	HEX	CHARACTER	DEC:	HEX	CHARACTER	DEC:	HEX	CHARACTER
064	40	@	080	50	P	096	60	'	112	70	p
065	41	A	081	51	Q	097	61	a	113	71	q
066	42	B	082	52	R	098	62	b	114	72	r
067	43	C	083	53	S	099	63	c	115	73	s
068	44	D	084	54	T	100	64	d	116	74	t
069	45	E	085	55	U	101	65	e	117	75	u
070	46	F	086	56	V	102	66	f	118	76	v
071	47	G	087	57	W	103	67	g	119	77	w
072	48	H	088	58	X	104	68	h	120	78	x
073	49	I	089	59	Y	105	69	i	121	79	y
074	4A	J	090	5A	Z	106	6A	j	122	7A	z
075	4B	K	091	5B	[107	6B	k	123	7B	{
076	4C	L	092	5C	\	108	6C	l	124	7C	
077	4D	M	093	5D]	109	6D	m	125	7D	}
078	4E	N	094	5E	^	110	6E	n	126	7E	~
079	4F	O	095	5F	_	111	6F	o	127	7F	△ (DEL)

Tab. A-1 Extended ASCII Character Set (cont.)

ASCII DISPLAY AND KEYBOARD CODE TABLES

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
128	80	Ç	144	90	É	160	A0	á	176	B0	▒
129	81	ü	145	91	æ	161	A1	í	177	B1	▒
130	82	é	146	92	Æ	162	A2	ó	178	B2	▒
131	83	â	147	93	ô	163	A3	ú	179	B3	
132	84	ä	148	94	Ö	164	A4	ñ	180	B4	┐
133	85	à	149	95	ò	165	A5	Ñ	181	B5	≡
134	86	å	150	96	û	166	A6	ä	182	B6	≡
135	87	ç	151	97	ù	167	A7	ö	183	B7	≡
136	88	ê	152	98	ÿ	168	A8	ı	184	B8	≡
137	89	ë	153	99	Ö	169	A9	┐	185	B9	≡
138	8A	è	154	9A	Ü	170	AA	┐	186	BA	
139	8B	ï	155	9B	¢	171	AB	1/2	187	BB	┐
140	8C	î	156	9C	£	172	AC	1/4	188	BC	┐
141	8D	ì	157	9D	¥	173	AD	ı	189	BD	┐
142	8E	Ä	158	9E	Pt	174	AE	«	190	BE	┐
143	8F	Å	159	9F	f	175	AF	»	191	BF	┐

Tab. A-1 Extended ASCII Character Set (cont.)

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
192	C0	┐	208	D0	└	224	E0	α	240	F0	≡
193	C1	┌	209	D1	┐	225	E1	β	241	F1	±
194	C2	└	210	D2	┌	226	E2	Γ	242	F2	≥
195	C3	┐	211	D3	└	227	E3	π	243	F3	≤
196	C4	—	212	D4	┐	228	E4	Σ	244	F4	∫
197	C5	+	213	D5	┐	229	E5	σ	245	F5	ℓ
198	C6	┐	214	D6	┌	230	E6	μ	246	F6	÷
199	C7	┐	215	D7	┐	231	E7	τ	247	F7	≈
200	C8	┐	216	D8	≠	232	E8	φ	248	F8	°
201	C9	┐	217	D9	┐	233	E9	⊖	249	F9	•
202	CA	┐	218	DA	┐	234	EA	Ω	250	FA	•
203	CB	┐	219	DB	■	235	EB	δ	251	FB	√
204	CC	┐	220	DC	■	236	EC	∞	252	FC	η
205	CD	=	221	DD	■	237	ED	∅	253	FD	₂
206	CE	┐	222	DE	■	238	EE	∈	254	FE	■
207	CF	┐	223	DF	■	239	EF	∩	255	FF	(SPACE) (SP)

Tab. A-1 Extended ASCII Character Set (cont.)

NATIONAL VARIATIONS IN EXTENDED ASCII CODE (ISO 8-bit code)

For Denmark, Norway, Greece and Portugal certain characters are displayed differently. These characters and their decimal and hexadecimal codes are shown in the following table.

ASCII DISPLAY AND KEYBOARD CODE TABLES

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
128	80	Ç	144	90	É	160	A0	á	176	B0	▒
129	81	ü	145	91	æ	161	A1	í	177	B1	▒
130	82	é	146	92	Æ	162	A2	ó	178	B2	▒
131	83	â	147	93	ô	163	A3	ú	179	B3	
132	84	ä	148	94	Ö	164	A4	ñ	180	B4	⌈
133	85	à	149	95	ò	165	A5	Ñ	181	B5	⌋
134	86	â	150	96	û	166	A6	ō	182	B6	⌈⌋
135	87	ç	151	97	ù	167	A7	Õ	183	B7	⌋
136	88	ê	152	98	ÿ	168	A8	ı	184	B8	⌋
137	89	ë	153	99	Ö	169	A9	ã	185	B9	⌋
138	8A	è	154	9A	Ü	170	AA	Ã	186	BA	⌋
139	8B	ï	155	9B	ø	171	AB	ℓ	187	BB	⌋
140	8C	î	156	9C	£	172	AC	'n	188	BC	⌋
141	8D	ì	157	9D	Ø	173	AD	ı	189	BD	⌋
142	8E	Ä	158	9E	Ł	174	AE	³	190	BE	⌋
143	8F	Å	159	9F	ı	175	AF	☒	191	BF	⌋

Tab. A-2 National Characters for Denmark and Norway

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
192	C0	┐	208	D0	┘	224	E0	α	240	F0	≡
193	C1	└	209	D1	≡	225	E1	β	241	F1	±
194	C2	┘	210	D2	≡	226	E2	Γ	242	F2	≥
195	C3	┘	211	D3	┘	227	E3	Π	243	F3	≤
196	C4	—	212	D4	┘	228	E4	Σ	244	F4	∫
197	C5	+	213	D5	≡	229	E5	σ	245	F5	J
198	C6	≡	214	D6	┘	230	E6	μ	246	F6	÷
199	C7	┘	215	D7	≡	231	E7	τ	247	F7	≈
200	C8	┘	216	D8	≡	232	E8	φ	248	F8	°
201	C9	≡	217	D9	┘	233	E9	⊖	249	F9	•
202	CA	┘	218	DA	┘	234	EA	Ω	250	FA	•
203	CB	≡	219	DB	■	235	EB	δ	251	FB	√
204	CC	┘	220	DC	■	236	EC	∞	252	FC	n
205	CD	=	221	DD	■	237	ED	∅	253	FD	₂
206	CE	≡	222	DE	■	238	EE	∈	254	FE	■
207	CF	┘	223	DF	■	239	EF	∩	255	FF	(SPACE) (SP)

Tab. A-2 National Characters for Denmark and Norway (cont.)

ASCII DISPLAY AND KEYBOARD CODE TABLES

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
128	80	A	144	90	P	160	A0	ι	176	B0	
129	81	B	145	91	Σ	161	A1	κ	177	B1	
130	82	Γ	146	92	T	162	A2	λ	178	B2	
131	83	Δ	147	93	Υ	163	A3	μ	179	B3	
132	84	E	148	94	Φ	164	A4	ν	180	B4	┐
133	85	Z	149	95	X	165	A5	ξ	181	B5	≡
134	86	H	150	96	Ψ	166	A6	ο	182	B6	┐
135	87	Θ	151	97	Ω	167	A7	π	183	B7	┐
136	88	I	152	98	α	168	A8	ρ	184	B8	≡
137	89	K	153	99	β	169	A9	σ	185	B9	≡
138	8A	Λ	154	9A	γ	170	AA	ς	186	BA	
139	8B	M	155	9B	δ	171	AB	τ	187	BB	┐
140	8C	N	156	9C	ε	172	AC	υ	188	BC	┐┐
141	8D	Ξ	157	9D	ζ	173	AD	φ	189	BD	┐┐
142	8E	O	158	9E	η	174	AE	χ	190	BE	≡
143	8F	Π	159	9F	θ	175	AF	ψ	191	BF	┐

Tab. A-3 National Characters for Greece

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
192	C0	┐	208	D0	⌌	224	E0	ω	240	F0	Ω
193	C1	└	209	D1	⌍	225	E1	α	241	F1	±
194	C2	┘	210	D2	⌎	226	E2	é	242	F2	≥
195	C3	┑	211	D3	⌏	227	E3	ñ	243	F3	≤
196	C4	—	212	D4	⌐	228	E4	ï	244	F4	∫
197	C5	+	213	D5	⌑	229	E5	í	245	F5	ℑ
198	C6	⌑	214	D6	⌒	230	E6	ó	246	F6	÷
199	C7	⌒	215	D7	⌓	231	E7	ú	247	F7	≈
200	C8	⌓	216	D8	⌔	232	E8	ü	248	F8	°
201	C9	⌔	217	D9	⌕	233	E9	ώ	249	F9	£
202	CA	⌕	218	DA	⌖	234	EA	Ά	250	FA	•
203	CB	⌖	219	DB	■	235	EB	Έ	251	FB	√
204	CC	⌗	220	DC	▀	236	EC	Ή	252	FC	π
205	CD	=	221	DD	▁	237	ED	Ί	253	FD	²
206	CE	≡	222	DE	▂	238	EE	Ό	254	FE	▮
207	CF	⌞	223	DF	▃	239	EF	Ύ	255	FF	(SPACE) (SP)

Tab. A-3 National Characters for Greece (cont.)

ASCII DISPLAY AND KEYBOARD CODE TABLES

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
128	80	Ç	144	90	É	160	A0	á	176	B0	▒
129	81	ü	145	91	æ	161	A1	í	177	B1	▒
130	82	é	146	92	Æ	162	A2	ó	178	B2	▒
131	83	â	147	93	ô	163	A3	ú	179	B3	
132	84	ä	148	94	ö	164	A4	ñ	180	B4	┘
133	85	à	149	95	ò	165	A5	Ñ	181	B5	≡
134	86	å	150	96	û	166	A6	õ	182	B6	≡
135	87	ç	151	97	ù	167	A7	Õ	183	B7	┘
136	88	ê	152	98	ÿ	168	A8	ı	184	B8	≡
137	89	ë	153	99	Ö	169	A9	ă	185	B9	≡
138	8A	è	154	9A	Ü	170	AA	Ă	186	BA	
139	8B	ï	155	9B	Á	171	AB	Ú	187	BB	┘
140	8C	î	156	9C	£	172	AC	Í	188	BC	┘
141	8D	ì	157	9D	À	173	AD	ı	189	BD	┘
142	8E	Ä	158	9E	Ê	174	AE	³	190	BE	≡
143	8F	Å	159	9F	Ô	175	AF	Ó	191	BF	┘

Tab. A-4 National Characters for Portugal

DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER	DEC	HEX	CHARACTER
192	C0		208	D0		224	E0		240	F0	
193	C1		209	D1		225	E1		241	F1	
194	C2		210	D2		226	E2		242	F2	
195	C3		211	D3		227	E3		243	F3	
196	C4		212	D4		228	E4		244	F4	
197	C5		213	D5		229	E5		245	F5	
198	C6		214	D6		230	E6		246	F6	
199	C7		215	D7		231	E7		247	F7	
200	C8		216	D8		232	E8		248	F8	
201	C9		217	D9		233	E9		249	F9	
202	CA		218	DA		234	EA		250	FA	
203	CB		219	DB		235	EB		251	FB	
204	CC		220	DC		236	EC		252	FC	
205	CD		221	DD		237	ED		253	FD	
206	CE		222	DE		238	EE		254	FE	
207	CF		223	DF		239	EF		255	FF	(SPACE) (SP)

Tab. A-4 National Characters for Portugal (cont.)

EXTENDED KEYBOARD CODES (FOR USA KEYBOARDS)

Certain keys and key combinations do not produce ASCII code (one byte). Instead they produce two bytes; the first byte is always zero. This zero value indicates an extended keyboard code. The following table shows the value of the second byte, when these key(s) are pressed.

ASCII DISPLAY AND KEYBOARD CODE TABLES

DECIMAL	HEXADECIMAL	KEY(S)
3	03	(Nul) NUL
15	0F	Cursor Left
16	10	ALT and Q
17	11	ALT and W
18	12	ALT and E
19	13	ALT and R
20	14	ALT and T
21	15	ALT and Y
22	16	ALT and U
23	17	ALT and I
24	18	ALT and O
25	19	ALT and P
30	1E	ALT and A
31	1F	ALT and S
32	20	ALT and D
33	21	ALT and F
34	22	ALT and G
35	23	ALT and H
36	24	ALT and J
37	25	ALT and K
38	26	ALT and L
44	2C	ALT and Z
45	2D	ALT and X
46	2E	ALT and C
47	2F	ALT and V
48	30	ALT and B
49	31	ALT and N
50	32	ALT and M
59	3B	F1
60	3C	F2
61	3D	F3
62	3E	F4
63	3F	F5
64	40	F6
65	41	F7
66	42	F8
67	43	F9
68	44	F10
71	47	HOME

DECIMAL	HEXADECIMAL	KEY(S)
72	48	Cursor up
73	49	PGUP
75	4B	Cursor Left
77	4D	Cursor Right
79	4F	END
80	50	Cursor Down
81	51	PGDN
82	52	INS
83	53	DEL
84	54	SHIFT and F1
85	55	SHIFT and F2
86	56	SHIFT and F3
87	57	SHIFT and F4
88	58	SHIFT and F5
89	59	SHIFT and F6
90	5A	SHIFT and F7
91	5B	SHIFT and F8
92	5C	SHIFT and F9
93	5D	SHIFT and F10
94	5E	CTRL and F1
95	5F	CTRL and F2
96	60	CTRL and F3
97	61	CTRL and F4
98	62	CTRL and F5
99	63	CTRL and F6
100	64	CTRL and F7
101	65	CTRL and F8
102	66	CTRL and F9
103	67	CTRL and F10
104	68	ALT and F1
105	69	ALT and F2
106	6A	ALT and F3
107	6B	ALT and F4
108	6C	ALT and F5
109	6D	ALT and F6
110	6E	ALT and F7
111	6F	ALT and F8
112	70	ALT and F9
113	71	ALT and F10

ASCII DISPLAY AND KEYBOARD CODE TABLES

DECIMAL	HEXADECIMAL	KEY(S)
114	72	CTRL and PRTSC
115	73	CTRL and Cursor Left
116	74	CTRL and Cursor Right
117	75	CTRL and END
118	76	CTRL and PGDN
119	77	CTRL and HOME
120	78	ALT and 1
121	79	ALT and 2
122	7A	ALT and 3
123	7B	ALT and 4
124	7C	ALT and 5
125	7D	ALT and 6
126	7E	ALT and 7
127	7F	ALT and 8
128	80	ALT and 9
129	81	ALT and 0
130	82	ALT and -
131	83	ALT and =
132	84	CTRL and PGUP

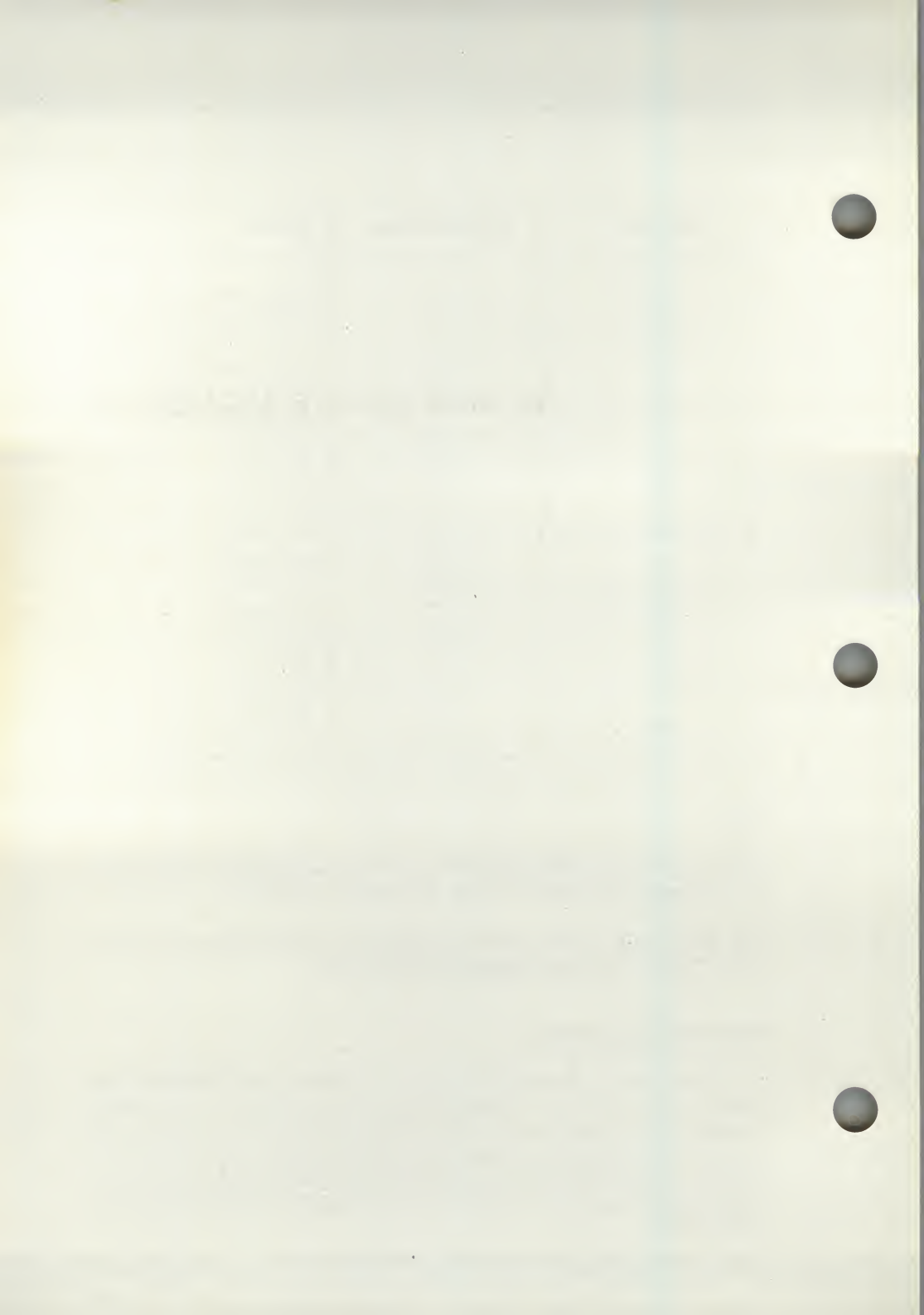
Note

Codes 120 to 131 decimal refer to the top row of keys in the body of the keyboard, not those on the right hand keypad.

The **SHIFT**, **CTRL** and **ALT** keys do not generate specific codes; they are used to modify the codes generated by other keys.

Note for Programmers

The above code is contained in the AH register, when BIOS INT 16H function 1 is executed. For extended keyboard codes 00 Hexadecimal is contained in the AL register. For displayable ASCII characters, normally the AH register contains the keyboard scan code and the AL register contains the ASCII code. See the MS-DOS System Programmer Guide Volume II for more details of National Keyboard Layouts and Codes.



B. ANSI ESCAPE SEQUENCES

ABOUT THIS APPENDIX

This appendix describes the ANSI escape sequences.

CONTENTS

INTRODUCTION	B-1
CURSOR MOVING FUNCTIONS	B-2
ERASING FUNCTIONS	B-4
GRAPHIC MODE FUNCTIONS	B-5
KEY REASSIGNMENT	B-7

ANSI ESCAPE SEQUENCES

INTRODUCTION

An ANSI escape sequence is a series of characters beginning with the character **ESC** (decimal 27, hex 1B) that can be generated to define functions to MS-DOS. Specifically you can affect cursor movement, erase specific areas of the screen and set the graphics mode, by issuing these sequences through standard MS-DOS input/output.

The sequences described in this appendix require the presence of the ANSI console driver. ANSI.SYS is a file included on your MS-DOS system diskette. To install the ANSI console driver the following command must be placed in the CONFIG.SYS file:

```
DEVICE = [d:][path] ANSI.SYS
```

Refer to Appendix C for information about the CONFIG.SYS file and the DEVICE command.

Remarks

The following notes are general to all escape sequences:

1. ESC can be generated in a variety of ways: changing your prompt (PROMPT = \$e); using the Video File Editor (see Chapter 6); or by program.
2. *Pn* represents a numeric parameter. This is a decimal number.
3. *Ps* represents a selective parameter. The parameter is still a decimal integer but is one that must be selected from a list of alternatives.
4. Where no parameter is specified, or where zero is entered, a default value is assumed.
5. No spaces should be typed inside escape sequences; any space shown in the specification is purely for ease of reading.

CURSOR MOVING FUNCTIONS

The following escape sequences affect the position of the cursor on the screen.

MNEMONIC	SEQUENCE	MEANING
CUP (Cursor Position) or HVP E (Horizontal and Vertical Position)	ESC [<i>Pn</i> ; <i>Pn</i> H or ESC [<i>Pn</i> ; <i>Pn</i> f	The cursor is moved to the line and column specified by the first and second parameters, respectively. The default values are 1. If no parameters are specified the cursor is moved to the home position.
CUU (Cursor Up)	ESC [<i>Pn</i> A	Moves the cursor up the screen by the number of rows specified by the parameter. If no parameter is specified one line is assumed. No action is taken if the cursor is already on the top line of the screen.
CUD (Cursor Down)	ESC [<i>Pn</i> B	Moves the cursor down the screen by the number of rows specified by the parameter. If no parameter is specified one line is assumed. No action is taken if the cursor is already on the bottom line of the screen.

ANSI ESCAPE SEQUENCES

MNEMONIC	SEQUENCE	MEANING
CUF (Cursor Forward)	ESC [Pn C	Moves the cursor to the right by the number of columns specified by the parameter. If no parameter is specified then one column is assumed. No action is taken if the cursor is already on the right-most column.
CUB (Cursor Backward)	ESC [Pn D	Moves the cursor left by the number of columns specified by the parameter. If no parameter is specified then one column is assumed. No action is taken if the cursor is already on the left-most column.
DSR (Device Status Report)	ESC [6 n	Causes the console driver to perform a CPR (Cursor Position Report) sequence.
CPR (Cursor Position Report)	ESC [Pn ; Pn R	The current cursor position is reported via the standard input/output device. The first parameter specifies the line and the second specifies the column. This sequence is performed by the console driver on receipt of a DSR sequence.

MNEMONIC	SEQUENCE	MEANING
SCP (Save Cursor Position)	ESC [s	The current cursor position is saved. The saved value can subsequently be restored by issuing an RCP (Restore Cursor Position) sequence.
RCP (Restore Cursor Position)	ESC [u	The current cursor position is restored to what it was at the time that the console Driver received the SCP sequence.

ERASING FUNCTIONS

The following sequences erase specific areas of the video display.

MNEMONIC	SEQUENCE	MEANING
ED (Erase Display)	ESC [2 J	The screen is erased and the cursor is moved to the home position.
EL (Erase Line)	ESC [K	The current line from the cursor position to the end of the line is erased, including the cursor position itself.

GRAPHIC MODE FUNCTIONS

MNEMONIC	SEQUENCE	MEANING
SGR (Set Graphics Rendition)	ESC [<i>Ps</i> ; ... ; <i>Ps m</i>	<p>The graphic functions specified by the parameters are invoked. The functions set by this sequence remain in effect until another SGR sequence is issued. The parameter values are as follows:</p> <ul style="list-style-type: none"> 0 - all attributes off (normal display) 1 - high intensity (bold) 5 - sets blink on 7 - reverse video on 8 - concealed on (makes display invisible) 30 - black foreground 31 - red foreground 32 - green foreground 33 - yellow foreground 34 - blue foreground on color display; underline on monochrome display 35 - magenta foreground 36 - cyan foreground

MNEMONIC	SEQUENCE	MEANING
		37 -white foreground 40 - black background 41 - red background 42 -green background 43 -yellow background 44 -blue background 45 -magenta background 46 -cyan background 47 -white background
SM (Set Mode)	ESC [= Ps h	<p>The screen width and type are set by the parameter specified. The possible parameter values are as follows:</p> 0 - 40 column by 25 line black and white 1 - 40 column by 25 line color 2 - 80 column by 25 line black and white 3 - 80 column by 25 line color 4 - 320 x 200 color

ANSI ESCAPE SEQUENCES

MNEMONIC	SEQUENCE	MEANING
		5 - 320 x 200 black and white 6 - 640 x 200 black and white 7 - automatic text wrap at end-of-line
RM (Reset Mode)	ESC [= Psl	Resets the attribute set by the SM sequence. Parameter values are the same as for the SM sequence.
Except for		7 - no text wrap at end-of-line (excess characters are not displayed)

KEY REASSIGNMENT

An ANSI escape sequence can be used to assign an ASCII code, a string, or a combination of ASCII codes and strings to any key or value or valid key-stroke combination.

Valid control sequences are:

ESC [*Pn* ; *Pn* ;... p

ESC [*Pn* ; "string" p

ESC [*Pn* ; "string"; *Pn*; *Pn*; "string"; *Pn* p

or any other combination of strings and decimal numbers.

The first parameter defines which ASCII code is to be mapped, unless it is zero, in which case the first and second parameters comprise the extended keyboard code. Refer to Appendix A for a complete list of ASCII and extended keyboard codes.

Note that the ASCII codes must be entered in decimal.

Examples

If your program issues the sequence...	THEN...
ESC[66;82p	pressing B becomes R instead of B.
ESC[0;67;"dir";13p	a string "dir" is entered followed by a carriage return whenever you press F9 . The initial 0 indicates that the F9 function key is represented by an extended keyboard code 67.

C. CONFIGURING MS-DOS

ABOUT THIS APPENDIX

This appendix describes how you can configure MS-DOS to suit the requirements of your application.

CONTENTS

INTRODUCTION	C-1
CONFIGURING COMMANDS	C-1
BREAK	C-1
BUFFERS	C-2
COUNTRY	C-3
DEVICE	C-5
DRIVPARM	C-6
FCBS	C-8
FILES	C-9
LASTDRIVE	C-10
SHELL	C-11
STACKS	C-12

INTRODUCTION

Whenever MS-DOS is initialized, it searches the root directory for a file named CONFIG.SYS. If this file exists, it is read and the configuring commands within it are executed. If CONFIG.SYS is not found the default setting for each command is used.

You can create or modify the CONFIG.SYS file using the Video File Editor or Line Editor, and include in it any of the commands described in this appendix.

Also if a particular command within CONFIG.SYS is not found its default setting is used.

CONFIGURING COMMANDS

BREAK



Sets or resets the **CTRL C** break facility.

BREAK = ON | OFF

Characteristics

The action of the **BREAK** command is exactly the same as described in Chapter 5. However, it can only occur once in the CONFIG.SYS file. The state set by including this command in the CONFIG.SYS file can later be reversed by entering a **BREAK** command with the opposite argument at the keyboard.

The default is **OFF**.

BUFFERS

Sets the number of buffers to be allocated.

BUFFERS = *number*

Where

SYNTAX ELEMENT	MEANING
<i>number</i>	<p>A decimal number in the range 1 to 99 that defines the number of buffers to be allocated.</p> <p>The default varies between 2 and 10, depending on the computer and its configuration.</p>

Characteristics

The optimum number of buffers depends on factors such as:

- the kind of disk drives attached to the computer
- whether external devices such as spooling tape are attached
- the type of applications most commonly used
- the amount of main memory
- whether the computer is acting as a network server

A minimum setting of **BUFFERS** = 5 is recommended for all computer configurations with disk drives, unless you have severe main memory constraints.

If your applications perform a lot of random accesses you will probably benefit by using a larger number of buffers. However, beyond a certain number it may take MS-DOS longer to scan the buffers than it would to access a value from disk, resulting in a drop in performance. For most data base applications between 10 and 20 buffers is a reasonable number.

Memory size affects the optimum number of buffers because each additional buffer requires 528 bytes of memory.

The optimum number of buffers can only be determined by trial then timing the result.

COUNTRY



Sets the country to allow MS-DOS to use the correct national time, date, currency and decimal separators.

COUNTRY = *number*

Where

SYNTAX ELEMENT	MEANING																																
<i>number</i>	<p>A three digit number which is the telephonic international country code:</p> <table><tr><td>001</td><td>United States of America</td></tr><tr><td>002</td><td>Canadian-French</td></tr><tr><td>031</td><td>Netherlands</td></tr><tr><td>032</td><td>Belgium</td></tr><tr><td>033</td><td>France</td></tr><tr><td>034</td><td>Spain</td></tr><tr><td>039</td><td>Italy</td></tr><tr><td>041</td><td>Switzerland</td></tr><tr><td>044</td><td>United Kingdom</td></tr><tr><td>045</td><td>Denmark</td></tr><tr><td>046</td><td>Sweden</td></tr><tr><td>047</td><td>Norway</td></tr><tr><td>049</td><td>Germany</td></tr><tr><td>061</td><td>Australia</td></tr><tr><td>351</td><td>Portugal</td></tr><tr><td>358</td><td>Finland</td></tr></table>	001	United States of America	002	Canadian-French	031	Netherlands	032	Belgium	033	France	034	Spain	039	Italy	041	Switzerland	044	United Kingdom	045	Denmark	046	Sweden	047	Norway	049	Germany	061	Australia	351	Portugal	358	Finland
001	United States of America																																
002	Canadian-French																																
031	Netherlands																																
032	Belgium																																
033	France																																
034	Spain																																
039	Italy																																
041	Switzerland																																
044	United Kingdom																																
045	Denmark																																
046	Sweden																																
047	Norway																																
049	Germany																																
061	Australia																																
351	Portugal																																
358	Finland																																

Note: If your country is not supported, choose the country which uses your national conventions.

The default is COUNTRY = 001

Installs a device driver.

DEVICE = *filespec*

Where

SYNTAX ELEMENT	MEANING
<i>filespec</i>	The file containing the device driver.

Characteristics

If *filespec* is the file ANSI.SYS, the escape sequences described in Appendix B are supported.

If *filespec* is DRIVER.SYS, refer to Appendix G for further details.

If *filespec* is VDISK.SYS, refer to Appendix F for further details.

Alternatively, you may enter the file name of any device driver written for your system.

Important

The device driver file must be in the root directory of the drive used for bootstrapping the computer.

Note: **DEVICE =** *filespec* can be repeated on several lines of the CONFIG.SYS file with different device driver parameters.

DRIVPARM

Enables you to override the default settings for predefined block devices.

DRIVPARM = */D: drive-no* [*/C*] [*/F: form-factor*] [*/H: head*] [*/N*] [*/S: sectors*] [*/T: tracks*]

Where

SWITCH	PARAMETER	MEANING
<i>/D</i>	<i>drive-no</i>	Specifies the logical drive number between 0 and 255. So 0 = A: 1 = B: 2 = C: etc.
<i>/C</i>		Disk change support is required. See "SHARE" in Chapter 5 for more details.
<i>/F</i>	<i>form-factor</i>	Specifies the media format and kind to be supported: 0 320/360 KB 1 1.2 MB 2 720 KB 3 8 inch single density 4 8 inch double density 5 Hard Disk 6 Tape Drive 7 Other

CONFIGURING MS-DOS

SWITCH	PARAMETER	MEANING
/H	<i>heads</i>	Specifies the number of heads on the disk drive. Its value can range from 1 to 99. The default is 2 heads.
/N		Specifies a non-removable block device such as a hard disk.
/S	<i>sectors</i>	Specifies the number of sectors per track. Its value can range from 1 to 99. The default is nine sectors per track.
/T	<i>tracks</i>	Specifies the number of tracks per side. Its value can range from 1 through 999. The default is 80 tracks per side.

Note

DRIVPARM can be repeated on several lines of the CONFIG.SYS file with different parameters.

Example

You might have a computer with an internal tape drive unit on drive "D:" that is configured at boot time to write 20 tracks of 40 sectors per track. If you want to reconfigure this tape drive to write 10 tracks of 99 sectors each, you can put the following line in your CONFIG.SYS file:

```
DRIVPARM = /D:3 /F:6 /H:1 /S:99 /T:10
```

This overrides the default device driver settings, and supports a tape drive as drive "D:" (in this case the logical and physical drive numbers are identical). This tape drive has 1 head, and supports a tape format of

10 tracks and 99 sectors per track. (This assumes that the device driver for the tape device supports this configuration of tracks and sectors). You might want to use this method to create a tape that you can read on another computer that can only read this alternate format.

See the "MS-DOS Software Installation Guide" for more details.

FCBS

Defines the number of FILES opened with File Control Blocks that can be open at any one time.

FCBS = *maxopen,number*

Where

SYNTAX ELEMENT	MEANING
<i>maxopen</i>	This is a number from 1 to 255 which represents the maximum number of files that can be opened with File Control Blocks. The default value is 4.
<i>number</i>	This is a number from 0 to 255 that specifies the files that MS-DOS cannot close automatically if the application tries to open more than maxopen. the first files opened are the protected files. The default value of number is 0.

Note: This command is only applicable when you are connected to a network. When not connected to a network, the number of files that can be open with File Control Blocks is 255.

FILES

Defines the number of file handles that can be open at any one time.

FILES = *number*

Where

SYNTAX ELEMENT	MEANING
<i>number</i>	<p>A decimal number in the range 1 to 255 defining the number of files that can be open concurrently.</p> <p>The default is 8.</p>

Characteristics

Each additional file above the default value of 8 requires an extra 48 bytes of memory. These 8 file handles include the 5 predefined file handles for: the console, standard input, standard output, auxiliary output and printer output.

The maximum number of file handles a program can have open is 20, including the 5 predefined file handles. When there are foreground and background processes, or in a multi-tasking environment, or on a server; each process can have 20 handles open. In these cases it is useful to set the number of handles larger than 20, the exact setting depends on the number of processes and the requirements of each process.

LASTDRIVE

Sets the maximum number of drives that you may access

LASTDRIVE = *drive-letter*

Where

SYNTAX ELEMENT	MEANING
<i>drive-letter</i>	Any letter from A through Z, defining the last valid drive that MS-DOS will accept.

Characteristics

The default is:

LASTDRIVE = **E**

In a network environment, you may use the NET USE command to assign the extra drive letters to a volume on a remote computer.

Remarks

You cannot set LASTDRIVE less than the number of drives on your computer.

Loads a top-level command processor, which can be COMMAND.COM or an alternative command processor.

SHELL = *pathname path* [/E: *nnnnn*] /P

Where

SYNTAX ELEMENT	MEANING
<i>pathname</i>	The drive, path and file name containing the top-level command processor to be loaded.
<i>path</i>	The drive and path of the directory containing the top-level command processor. This sets the environment variable COMSPEC to point to the command processor for reloading purposes.
/E: <i>nnnnn</i>	This switch specifies the environment size, where <i>nnnnn</i> is the size in bytes. The size may range between 160 and 32768 bytes. The default value is 160 bytes.
/P	This switch must be specified. This is to indicate that the command processor is to be used for system startup. If omitted AUTOEXEC.BAT will not be executed and an EXIT causes a system crash.


Example

```
SHELL = C:\BIN\COMMAND.COM C:\BIN /E:16384 /P
```

This command causes the top level command processor to be loaded from the BIN directory of the C: drive. The second parameter indicates that the transitory part of the command processor is to be reloaded from the BIN directory of the C: drive. The /E switch sets the environment size to be 16384 bytes. The /P switch must be specified to indicate this is the top level command processor.

Remarks

If you are writing an alternative command processor, remember to duplicate COMMAND.COM's internal commands, batch processor and program loader.



STACKS

Starting with MS-DOS Version 3.20 hardware interrupt stacking uses a separate stack, rather than the user stack as in previous versions. This command sets the number and size of stack frames available for hardware interrupt stacking.

STACKS = *frames,size*

CONFIGURING MS-DOS

Where

SYNTAX ELEMENT	MEANING
<i>frames</i>	Specifies the number of stack frames available. This number can range from 8 through 64. The default number is 9.
<i>size</i>	Specifies the size in bytes of each stack frame. This number can range from 32 through 512. The default is 128.

Characteristics

When a hardware interrupt occurs, MS-DOS allocates a stack frame from the available frames. When the BIOS has processed the interrupt, MS-DOS frees the stack frame. If too many hardware interrupts are queued for the available stack resources, the system crashes and outputs the following message:

FATAL: Internal Stack Failure, System Halted.

If this situation should occur on your personal computer, increase the number of stack frames. The default stack size of 128, is sufficient except in exceptional circumstances. The pool of stack frames is within the memory allocated at the initialization of MS-DOS: increasing the STACKS parameters reduces available memory by the corresponding number of bytes.

D. ERROR MESSAGES

ABOUT THIS APPENDIX

This appendix explains the various error messages that can be displayed by MS-DOS and its utilities.

CONTENTS

DEVICE AND DISK DRIVE ERRORS	D-1
ERROR MESSAGES IN ALPHABETICAL ORDER	D-3

ERROR MESSAGES

DEVICE AND DISK DRIVE ERRORS

Errors may occur when reading from or writing to devices and disk drives. These errors cause the system to stop and output a message of this form:

type error action unit
Abort, Retry, Ignore?

Where

type error Specifies the possible causes of device or disk drive failure. The Table D-1 contains the message type.

action Can be **reading** or **writing**.

unit Can be either:

device *device-name*

or

drive *drive-letter*

It specifies the device or disk drive in error.

device-name See the section "Reserved Device Names" in Chapter 3 for a list of device-names.

drive-letter A single letter in the range "A" through "Z".

Response

When you receive one of these messages, do one of the following:

Enter **A** for Abort. The system ends the program that requested the read or write.

Enter **R** for Retry. The system tries the read or write operation again.

Enter **I** for Ignore. The system ignores the error and attempts to continue the program. (This method may cause loss of data).

Important

For disk drive error messages, do not change disks before responding with **A**, **R** or **I**. The only exception is "Invalid Disk Change".

The following table lists possible causes of device and disk drive errors; these are described along with the rest of the error messages in the section "Error Messages In Alphabetical Order".

- Bad call format
- Bad command
- Bad unit
- Data
- Disk error
- FCB unavailable
- General Failure
- Invalid Disk Change
- Lock Violation
- No paper
- Non-DOS disk
- Not ready
- Read fault
- Sector not found
- Seek
- Sharing buffer exceeded
- Sharing Violation
- Write fault
- Write protect

Tab. D-1 Possible Causes of Device and Disk Drive Errors

Examples

The following example is a typical error message displayed when the printer connected to the computer is switched off:

**Not ready error writing device PRN
Abort, Retry, Ignore?**

ERROR MESSAGES

The following example is a typical error message displayed when there is no floppy diskette in the disk drive:

Not ready error reading drive A
Abort, Retry, Ignore?

ERROR MESSAGES IN ALPHABETICAL ORDER

The following list contains a description of the possible cause and meaning of the message and where possible suggests remedial action.

ERROR MESSAGE	MEANING
Abort edit (Y/N)? (EDLIN)	MS-DOS displays this message when you choose the Q (Quit) command in EDLIN. The Quit command exits the editing session without saving any editing changes. Specify Y (for "Yes") or N (for "No").
Access denied (MS-DOS)	MS-DOS displays this message when you tried to write to or delete a file marked as read only. If you really want to carry out this action, use the ATTRIB command to give the file a read/ write attribute.
All files canceled by operator (PRINT)	MS-DOS displays this message when you specify the /T switch with the PRINT command.
All partitions are currently in use (FDISK)	Self-explanatory.
All specified files are contiguous (CHKDSK)	All files are allocated contiguously on the disk without fragmentation.
Allocation error size adjusted (CHKDSK)	<p>The size of the file indicated in the directory was not consistent with the amount of data actually allocated to the file.</p> <p>Adjustment actually takes place only, if you specify the /F switch with CHKDSK, the file is truncated at the end of the last valid cluster.</p>

ERROR MESSAGES

ERROR MESSAGE	MEANING
Ambiguous switch:z (LINK)	The characters in z identify more than one linker parameter.
Amount read less than size in header (EXE2BIN)	The file is smaller than its header indicates. Recompile (or reassemble) and relink the program.
An internal failure has occurred (LINK)	The linker program has failed. Report the conditions of the failure to your Olivetti Dealer.
Attempt to access data outside of segment bounds, possibly bad object module (LINK)	An invalid object file has been specified.
Attempt to write on write-protected diskette (FORMAT)	You cannot format a write-protected diskette. Use another disk or remove the write-protection tag.
Backup file sequence error (RESTORE)	The file being restored is backed up on more than one diskette. The wrong diskette in the sequence has been inserted.
Bad call format (device error)	A request header of incorrect length was passed to a device driver. Contact your Olivetti dealer.

ERROR MESSAGE	MEANING
Bad command (device error)	A device driver issued an incorrect command to the device specified in the error message.
Bad command or file name (MS-DOS)	You typed neither an internal command nor an external command (executable filename). You either mistyped the command or the file does not exist in the specified disk directories.
Bad numeric parameter (LINK)	The value specified with the /STACK parameter is not a valid numeric constant.
Bad or missing Command Interpreter (MS-DOS)	The MS-DOS disk being loaded does not contain the file COMMAND.COM in the root directory. This message may also appear if an error occurs during loading of the system disk or if the COMSPEC = <i>parameter</i> does not point to a directory containing COMMAND.COM.
Bad or missing filename (MS-DOS)	One of the following conditions occurred during startup: The device driver named in the DEVICE = <i>parameter</i> does not exist in CONFIG.SYS. A break address has been set which is out of bounds for the machine. An error occurred during loading of the driver.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Bad Partition Table (FORMAT)	You have either tried to FORMAT a hard disk partition which does not exist or the partition's partition table is bad or invalid. If there is an existing partition, delete it using FDISK. Create a new partition and try formatting it again.
Bad unit error (device error)	An invalid subunit number has been sent to a device driver. Contact your dealer.
Batch file missing (MS-DOS)	A file being processed in batch mode is no longer present. It may have been removed or erased during processing. Batch processing stops and control returns to MS-DOS.
xxxxx bytes in bad sectors (CHKDSK, FORMAT)	Bad clusters have been marked in the File Access Table (FAT). A cluster consists of one or more sectors. This stops the bad sectors on the disk from being used by MS-DOS for files. On a hard disk it is a normal occurrence for there to be a small percentage of bad sectors. If the percentage grows too large on a hard disk call your service engineer for advice.
bf Error (DEBUG)	The specified flag code setting is invalid. Re-enter the Register (RF) command with the correct code.

ERROR MESSAGE	MEANING
bp Error (DEBUG)	More than ten breakpoints were specified for the Go (G) command. Re-enter the GO command with ten or fewer breakpoints.
br Error (DEBUG)	An invalid register name has been specified. Re-enter the Register (R) command with a valid register name.
Cannot CHDIR to root Processing cannot continue (CHKDSK)	The disk you are checking is faulty. Reboot MS-DOS and try to RECOVER the disk.
Cannot CHKDSK a Network drive (CHKDSK)	You cannot check drives which are redirected over the Network.
Cannot CHKDSK a SUBSTed or ASSIGNED drive (CHKDSK)	You cannot check drives which are SUBSTed or ASSIGNED.
Cannot COPY <i>filename</i> to from a reserved device (XCOPY)	You cannot XCOPY files to or from a reserved device, such as CON: or PRN:.
Cannot do binary reads from a device (COPY)	You have tried to use the /B switch with the name of a device. Place a /A switch after the device name to copy in ASCII mode.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Cannot edit .BAK file rename file (EDLIN)	You attempted to edit a backup copy created by EDLIN. Either rename the file or copy the .BAK file and give it a different extension.
Cannot execute (SELECT)	An error occurred when SELECT tried to call the FORMAT or XCOPY commands: make sure these commands are present on the Source Drive and re-enter the SELECT command.
Cannot find file object file. Change diskette hit ENTER (LINK)	The specified object module is not present on the diskette.
Cannot find library file. ENTER new drive letter: (LINK)	The specified library is not present on the current drive. Enter the drive containing the library.
Cannot format an ASSIGNED or SUBSTed drive (FORMAT)	You attempted to format a drive which is actually mapped to another drive by the ASSIGN or SUBST command. Run ASSIGN or SUBST again and clear all drive mappings.
Cannot FORMAT a Network drive (FORMAT)	You cannot format drives that are redirected over the Network.

ERROR MESSAGE	MEANING
Cannot load COMMAND, system halted (MS-DOS)	<p>One of the following conditions occurred while loading the command processor:</p> <ul style="list-style-type: none"> • The available memory map has been destroyed. • The command processor specified by the COMSPEC parameter does not exist. • Reboot MS-DOS.
Cannot nest response file (LINK)	It is not possible to use an @ filespec within an automatic response file.
Cannot open filename (PRINT)	Either MS-DOS cannot find the specified file to print or the file does not exist. Check the command for a valid filename.
Cannot open <i>filespec1 filespec2</i> (FC)	Either FC cannot find the specified file(s) or the file(s) do not exist. Check the command for a valid filename.
Cannot open list file (LINK)	A list file cannot be opened because the disk or directory is full.
Cannot open overlay (LINK)	An overlay cannot be opened because the disk or directory is full.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Cannot open response file (LINK)	The specified response file does not exist.
Cannot open temporary file (LINK)	The directory or disk is full, hence the linker cannot create the VM.TMP file. Insert a new disk. Do not remove the disk that will receive the List.MAP file.
Cannot perform a cyclic copy (XCOPY)	You have used the /S switch and have specified a target directory, which is a subdirectory of the source.
Cannot recover . entry, processing continued (CHKDSK)	The . entry (working directory) is defective.
Cannot recover .. entry, processing continued (CHDSK)	The .. entry (parent directory) is defective.
Cannot RECOVER a Network drive (RECOVER)	You cannot recover files on drives that are redirected over the Network.
CHDIR .. failed, trying alternate method (CHKDSK)	In traveling the tree structure, CHKDSK was not able to return to a parent directory. It will try to return to that directory by starting over at the root and traveling down.
COM port does not exist (MODE)	You have specified an invalid "COM" port

ERROR MESSAGE	MEANING
Compare Error(s) (DISKCOMP)	Different information has been found on one or more disk locations.
Compare error at offset XXXXXXX (COMP)	While comparing two files, different values were found at offset XXXXXXXX (hexadecimal). The values found are also displayed (in hexadecimal).
Compare more files (Y/N)? (COMP)	Answer Y if you wish to compare more files, otherwise enter N .
Contains XXX noncontiguous blocks (CHKDSK)	This message indicates that your files are fragmented. Fragmented files take longer to read COPY badly fragmented files to a newly formatted disk. Using the new disk will result in faster reading of the files.
Content of destination lost before copy (MS-DOS)	A file to be used as a source file to the Copy command has been overwritten prior to completion of the copy. Example: COPY F1 + F2 F2 destroys F2 before it can be copied.
Current drive is no longer valid > (COMMAND.COM)	If you have set your prompt to include the \$p parameter and have specified an invalid drive you will first get the message: "Not ready error reading drive drive-letter" Abort, Retry, Ignore? Press I in response and the Current drive message is displayed. Type in a valid <i>drive-letter</i> : to change the current drive to a valid drive.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Data error (device error)	Data could not be read/written correctly because of a faulty disk.
DEVICE Support Not Present (DISKCOMP) (DISKCOPY)	One of the floppy disk drives specified in the DISKCOMP command does not support MS-DOS Ver. 3.2 device control.
df Error (DEBUG)	Conflicting codes have been specified for a single flag. A flag can be changed only once for each Register (RF) command.
Directory entries adjusted (VDISK)	VDISK has adjusted the number of directory entries in the parameters of DEVICE=VDISK.SYS in the CONFIG.SYS command.
Directory error in TREE (TREE)	Self-explanatory.
Directory is joined, tree past this point not processed. (CHKDSK)	CHKDSK will not process directories which are joined.
Directory is totally empty, no . or ..., tree past this point not processed. (CHKDSK)	The specified directory does not contain references to working and parent directories. Delete the specified directory and recreate it.
Directory not empty (JOIN)	You can only JOIN onto a directory which is empty.

ERROR MESSAGE	MEANING
Disk already has an MS-DOS partition (FDISK)	You cannot use the "Create DOS Partition" option on a fixed disk which already has a DOS partition.
Disk error (device error)	An error has occurred reading from or writing to a disk.
Disk error reading FAT <i>copy</i> (CHKDSK)	An error occurred while CHKDSK was trying to read the file allocation table <i>copy</i> has the value 1 or 2.
Disk error writing FAT <i>copy</i> (CHKDSK)	An error occurred while CHKDSK was trying to update the file allocation table <i>copy</i> has the value 1 or 2.
Disk full. Edits lost (EDLIN)	EDLIN was not able to save your file due to lack of disk space.
Disk unsuitable for system disk (FORMAT)	The diskette contains a defective track where MS-DOS files must reside. The disk may only be used for data.
Diskette bad or incompatible (COPY)	You will get this message, if there is a read error from your source diskette or if there is a write error to your target diskette.
Divide overflow (MS-DOS)	A divide by zero was attempted, or an internal logic error has occurred. The system continues as if CTRL BREAK had occurred.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Drive types or diskette types not compatible (DISKCOPY, DISKCOMP)	The source and target diskettes must have the same format capacity. See the respective command specifications.
(.)(..) Does not exist (CHKDSK)	This message indicates either the . or .. directory entry is invalid.
Duplicate filename or file not found (RENAME)	Either an attempt has been made to rename a file with a file name that already exists in the directory, or the file to be renamed could not be found on the specified (or default) drive.
Dup record too complex (LINK)	There is a problem with an object module created from an assembler source program. A single DUP requires 1024 bytes before expansion. Debug the source program then return to the linker.
End of input file (EDLIN)	The entire file was read into memory. If the file is read in sections, this message indicates the last section of the file is in memory.
Entry error (EDLIN)	You have entered an EDLIN in command correctly. Re-enter the command.
Entry has a bad link/attribute/size (CHKDSK)	This message may be preceded by one or two periods indicating which subdirectory is invalid. If you have specified the /F switch, CHKDSK will try to correct the error.

ERROR MESSAGE	MEANING
Eof mark not found (COMP)	The end of valid data in the last block of two files being compared has not been found. Most likely to occur in non-test files.
Error in EXE file (MS-DOS)	The file contains erroneous relocation information created by LINK. The file may have been altered after creation.
Error in EXE or HEX file (DEBUG)	The EXE or HEX file contained invalid characters or records.
Error in IOCTL call (FORMAT)	You are trying to format a device, that does not need formatting.
ERROR Incorrect DOS version (DRIVER.SYS)	You are not boot-strapping MS-DOS Ver. 3.20. The device DRIVER.SYS will only work with MS-DOS Ver. 3.20.
ERROR No Drive Specified (DRIVER.SYS)	You did not specify a drive number, when you declared DRIVER.SYS in your CONFIG.SYS.
Error loading system from fixed disk (FDISK)	The operating system cannot be loaded from the fixed disk. Retry, or if that fails, boot the system from diskette and put a new copy of MS-DOS onto the fixed disk using the SYS command.
Error reading drive x (RECOVER)	Self-explanatory.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Error reading file (PRINT)	Self-explanatory.
Error reading partition table (FDISK)	Five unsuccessful attempts have been made to read the startup record from hard disk. Retry FDISK, or if that fails, try running your Customer Test as instructed by your Installation and Operations Guide.
Error writing to device (COMMANDS)	The device issued an I/O error and your data was not written. Retry.
Error writing partition table (FDISK)	Five unsuccessful attempts, have been made to write the startup record on the fixed disk. Retry FDISK, or if that fails, try running your Customer test as instructed by your Installation and Operations Guide.
Errors found, F parameter not specified. Corrections will not be written to disk (CHKDSK)	As the /F parameter was not used, an analysis of the disk will be made and the results displayed, but no corrections will be written to the disk.
Errors on list device indicate that it may be offline. Please check it. (PRINT)	Your printer is off-line.

ERROR MESSAGE	MEANING
EXEC failure (MS-DOS)	One of the following conditions occurred while reading a file from disk: Read error occurred. The FILES = <i>parameter</i> in the configuration file is not large enough. Increase the value and restart MS-DOS.
EXE and HEX files cannot be written (DEBUG)	The data would require a backward conversion that DEBUG does not support.
File allocation table bad drive x Abort, Retry, Ignore? (MS-DOS, CHKDSK)	See "Device and Disk Drive Errors" at the beginning of this appendix. If the error persists, the disk should be reformatted.
Files are different sizes (COMP)	The specified files are not of the same length and cannot be compared.
File xxx canceled by operator (PRINT)	When the operator cancels printing, this message appears on the screen.
File cannot be converted (EXE2BIN)	The input file you have specified does not have the correct format for conversion.
File cannot be copied into itself (COPY)	A request was made to COPY a file and place the copy (with the same name) in the same directory as the source file. Either change the name given to the copy or put it on another diskette or directory.

ERROR MESSAGES

ERROR MESSAGE	MEANING
File Error (SELECT)	You have a problem with your source or target diskette. Run CHKDSK on them to determine the cause of the error.
File creation error (MS-DOS and commands)	An unsuccessful attempt was made to add a new file to the directory. Run CHKDSK to determine the cause of the error.
File is READ-ONLY (EDLIN)	You may not change this file because the file is designated read-only. If you really want to write to this file, use the ATTRIB command to give the file a read/write attribute.
File name must be specified (EDLIN)	You did not specify a file name when you started EDLIN.
File not found (COMMANDS)	The file you named in the Transfer command does not exist.
File not found (MS-DOS and commands)	A file named in a command does not exist on the disk in the specified (or default) drive.
File not in PRINT queue (PRINT)	The file you want to remove from the print queue is not in the queue.
FIND: File not found (FIND)	A non-existent file name was specified when issuing a FIND command.

ERROR MESSAGE	MEANING
FIND: Invalid number of parameters (FIND)	A string was not specified when issuing a FIND command.
FIND: Invalid parameter <i>option-name</i> (FIND)	You specified an invalid parameter to the FIND command.
FIND: Read error in <i>filename</i> (FIND)	An error occurred when FIND tried to read the file specified in the command.
FIND: Syntax error (FIND)	You entered an illegal string when issuing the FIND command.
First cluster number is invalid, file truncated (CHKDSK)	An invalid pointer to the data area has been found in the file whose name precedes this message. If /F was specified, the file is truncated to zero length.
Fixup offset exceeds field width (LINK)	An assembly language instruction refers to an address with a short instruction instead of a long instruction. Edit the assembler source program and process it again.
Fixups needed-base segment (hex): (EXE2BIN)	The source (.EXE) file contained information indicating that a load segment is required for the file. Specify the absolute segment address at which the finished module is to be located.

ERROR MESSAGES

ERROR MESSAGE	MEANING
FOR cannot be nested (Batch)	It is not possible to have more than one FOR subcommand on one command line.
Format broken. (FORMAT)	Self-explanatory.
Format failure (FORMAT)	A disk error occurred on the diskette being formatted. The diskette cannot be formatted.
Format not supported on drive <i>drive</i> : (FORMAT)	If you have tried to format a virtual disk, you will have received this error message. You do not need to format a virtual disk. If you are using a block device declaration of DEVICE or DRIV- PARM in your CONFIG.SYS, try changing the parameters or removing the declaration. If you have no device declaration for that drive in your CONFIG.SYS and have received this error message, try reinstalling MS-DOS.
Formatting target while copying (DISKCOPY)	Unformatted tracks have been found on the target diskette. These tracks will be formatted as copying proceeds.

ERROR MESSAGE	MEANING
General Failure (device error)	This message is displayed when no other device error message is suitable. Check the device or disk drive referred to in the message. If you have found and corrected the cause of the error press R for Retry, otherwise press A for abort.
Has invalid cluster, file truncated (CHKDSK)	The specified file contains an invalid data area. If the /F parameter was used, the file is truncated at the last valid data block.
Illegal device name (MODE)	<p>Permitted device names are:</p> <p>Printers LPT1, LPT2, LPT3.</p> <p>The Asynchronous Communications Adapter must be either COM1 or COM2.</p> <p>Only one space is allowed between MODE and its parameter(s).</p>
Illegal parameter (MODE)	Self-explanatory.
Incorrect DOS version (Various EXTERNAL MS-DOS COMMANDS)	Many DOS utilities will not run on older versions of MS-DOS. For example the utilities CHKDSK, PRINT, and SYS will only run under the exact version of MS-DOS with which they were distributed.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Incorrect number of parameters (JOIN) (SUBST)	You specified too many or too few options in the command line.
Incorrect parameter (ASSIGN) (SHARE)	One of the options you specified is wrong.
Incorrect Version of MODE (MODE)	Use the version from the MS-DOS System disk you used to install your operating system.
Input file read error (LINK)	An invalid object file has been entered in the command line.
Insufficient disk space (MS-DOS and commands)	The disk does not contain enough free space to contain the new file.
Insufficient memory (COMMANDS)	There is not enough memory to perform the specified operation.
Insufficient memory for system transfer (FORMAT)	Your memory configuration is insufficient to transfer the MS-DOS hidden system files, when you specified the /S switch.

ERROR MESSAGE	MEANING
Insufficient room in root directory. Erase files in root and repeat CHKDSK (CHKDSK)	CHKDSK cannot create an entry in the root directory for saving lost chains as files (see message "X lost clusters found in Y chains -". Convert lost chains to files (Y/N?)" because the root directory is full. You should copy some files from the root directory to another disk, then reexecute CHKDSK.
Insufficient space on disk (DEBUG)	A write command was issued to a disk that does not have enough free space to hold the data being written.
Intermediate file error during pipe (MS-DOS)	<p>One of the following errors has occurred:</p> <ul style="list-style-type: none"> • One or both intermediate MS-DOS files cannot be created because the root directory is full. • The piping files do not exist. • There is insufficient space on the disk for the data being piped. <p>Remove some files from the root directory and retry. If this fails, a piping file has been erased. Correct the program and retry.</p>
INTERNAL ERROR in MODE application (MODE)	If this error occurs, report the circumstances to your dealer.
Invalid baud rate specified (MODE)	Self explanatory

ERROR MESSAGES

ERROR MESSAGE	MEANING
Invalid characters in volume label (FORMAT)	Volume labels may contain up to 11 printable characters without a period (.).
Invalid COMMAND.COM in drive X (MS-DOS)	<p>The program you have just run used up almost all of memory. MS-DOS must now reload the transient part of COMMAND.COM file from disk. However, MS-DOS cannot find COMMAND.COM on the disk or the copy found is invalid. Insert a disk into the X: drive which contains the same version of COMMAND.COM that you started with.</p> <p>Press any key to commence the reloading.</p>
Invalid country code (MS-DOS) (SELECT)	You have specified a country number in your SELECT command or CONFIG.SYS file which is not configured in this implementation of MS-DOS. Country codes must be in the range 1-99 and are the same as the International dialing code for the selected country.
Invalid current directory (MS-DOS)	Your disk is bad. Replace the disk or make another copy from your backup system disk.
Invalid current directory. Processing cannot continue (CHKDSK)	CHKDSK has found an error in the disk's current directory. Restart the system and rerun CHKDSK.

ERROR MESSAGE	MEANING
Invalid date (MS-DOS, DATE)	You specified an invalid date in response to the date prompt when starting MS-DOS or when using the DATE command.
Invalid device (CTTY)	The specified device name is invalid in MS-DOS.
Invalid device parameters from device driver (FORMAT)	FORMAT displays this message when the number of hidden sectors is not evenly divisible by the number of sectors per track (i.e., the partition does not start on a track boundary). This might happen if you tried to format a hard disk that previously had been formatted with MS-DOS Version 2.11 without first running FDISK.
Invalid directory (MS-DOS)	The directory you specified either does not exist or is invalid. Check to see that you entered the directory name correctly.
Invalid disk change	You changed the disk in a drive when it was not allowed. Put the disk back in the drive and press R for Retry.
Invalid drive in search path (MS-DOS)	One of the paths specified in the PATH command contains an invalid drive name. This error occurs during execution, not during the PATH command.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Invalid drive or filename (EDLIN.FC)	You did not specify a valid drive or file name when invoking EDLIN or FC.
Invalid drive specification (MS-DOS and commands)	You have tried to enter an invalid drive specifier in a command line.
Invalid drive was specified (MS-DOS and commands)	Self-explanatory.
Invalid environment (MS-DOS)	The environment contains invalid characters.
Invalid environment size specified (MS-DOS)	Either you specified an invalid parameter to the environment switch "/E:size" or you specified size to be a number less than 160 or greater than 32768
Invalid format (LINK)	An error has been found in a library.
Invalid handle (MS-DOS)	A program has attempted to access a file using an an invalid file handle.
Invalid media or Track 0 bad-disk unusable (FORMAT)	You cannot use the disk you are trying to format because track 0 is damaged or the disk is of the wrong type.
Invalid memory block address (MS-DOS)	A program has specified an invalid memory block address.

ERROR MESSAGE	MEANING
Invalid number of parameters (COMMANDS)	You have specified the wrong number of parameters on the command line. Check the syntax of the command you are using.
Invalid numeric parameters (LINK)	A character other than a digit has been included in a numeric parameter.
Invalid numeric switch specification switch error: "s:XXX" (LINK)	You specified a character for a switch requiring a numeric parameter. LINK will abort.
Invalid object module (LINK)	Object module(s) are incorrectly formed or incomplete
Invalid parameter (Commands)	The parameter entered for a command was incorrect. Check the syntax of the command you are using.
Invalid parameters (MODE)	<p>This message indicates one of:</p> <ul style="list-style-type: none"> • No parameters entered. • First letter not L or C. • First parameter not one of: 40, 80, BW40, BW80, CO40, CO80, MONO, L, R. • The referenced display adapter is not present.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Invalid partition table on fixed disk (FDISK)	Invalid partition information has been detected during startup from fixed disk. Restart DOS using diskette and correct the fixed disk partition information using the FDISK command.
Invalid path (TREE)	A directory whose name is in another directory cannot be accessed by TREE. Use CHKDSK to determine the error in the directory structure.
Invalid path, not directory, or directory not empty (RMDIR)	The directory cannot be removed as the path contains an invalid name, or the directory is not empty. The current directory cannot be removed.
Invalid path, or filename (COPY)	A directory or filename that does not exist has been specified.
Invalid subdirectory entry (CHKDSK)	The subdirectory whose name precedes this message contains invalid information. For more detailed information rerun CHKDSK with the /V switch.
Invalid switch: z (LINK)	The characters in z do not identify a valid linker parameter.
Invalid switch specification (FORMAT)	Self-explanatory.

ERROR MESSAGE	MEANING
Invalid time (MS-DOS) (TIME)	You specified an invalid time in response to the time prompt when starting MS-DOS or using the TIME command.
Invalid Volume ID (FORMAT)	When you tried to format an existing hard disk partition and were prompted for the existing volume label, you entered an incorrect volume label. If you really intend to format that hard disk partition, use the VOL command to find out the correct volume label, and re-enter the FORMAT or SELECT command.
<i>filename</i> is cross linked on cluster (CHKDSK)	You have two files cross linked. Make a copy of the file you want to keep, and then delete both files that are cross linked.
Keyboard routine not found (SELECT)	You indicated that the keyboard routine "KEYBxx" was to be found on the source diskette. It was not found.
Label not found (Batch)	A non-existent label has been specified in a GOTO command within the batch file.
Last backup diskette not inserted (BACKUP, RESTORE)	The diskette in the drive is not the last one in a series created by BACKUP.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Line too long (EDLIN)	During a Replace (R) command, the string given as the replacement caused the line to expand beyond the 253-character limit. The Replace command is ended abnormally. Split the long line into shorter lines; then reissue the Replace command.
Lock Violation (device error)	A program tried to access part of a file that is being used by another program. Press A for Abort or wait a while and press R for Retry.
XXX lost clusters found in YYY chains Convert lost chains to files (Y/N)? (CHKDSK)	If you respond Y to this prompt and had specified the /F switch, CHKDSK will recover the lost blocks it found when checking the disk. CHKDSK will create a directory entry and a file for you with the filename FILEnnnn.CHK. If you respond N and had specified /F switch, CHKDSK frees the lost blocks so they can be reallocated.
Memory allocation error. Cannot load COMMAND, system halted (MS-DOS)	The available memory map has been destroyed. You must restart MS-DOS.
Memory control blocks destroyed (MS-DOS)	Self-explanatory.
Missing country code (SELECT)	When you entered the SELECT command, you forgot the country code parameter.

ERROR MESSAGE	MEANING
Missing keyboard code (SELECT)	When you entered the SELECT command, you forgot the keyboard code parameter.
MS-DOS incorrect on default drive (Startup)	Self-explanatory.
MS-DOS not found on default drive (Startup)	Self-explanatory.
Must specify destination line number (EDLIN)	A destination line number was not specified for a Copy or Move command. Reenter the command correctly.
New file (EDLIN)	This message is printed if EDLIN does not find a file with the name you specified if you are creating a new file, ignore this message. If you do not intend to create a new file, check to see that you correctly typed the filename of the file you wish to edit.
No COM: ports	You have no "COM" ports attached to your computer
No files added (REPLACE)	You specified the /A switch, however all the files on the source exist on the target.

ERROR MESSAGES

ERROR MESSAGE	MEANING
No files replaced (REPLACE)	No files on the target were replaced.
No fixed disks present (FDISK)	<p>You cannot run the FDISK program due to one of the following conditions:</p> <ul style="list-style-type: none">• No fixed disk attached.• Fixed disk is present in the expansion unit but the unit is not switched on.• Fixed disk is incorrectly installed.
No free file handles Cannot start COMMAND exiting (MS-DOS)	Reload MS-DOS. If this message persists, increase the size of the FILES = <i>parameter</i> in the CONFIG.SYS file, and reload MS-DOS.
No free file handles (MS-DOS)	You have tried to load a second copy of the command processor, but too many files are currently open. Increase the size of the FILES = <i>parameter</i> in the configuration file and reload MS-DOS.
No MS-DOS partition to delete (FDISK)	You have used the Delete DOS Partition option when no such partition exists on the current fixed disk or you have tried to delete a non-existent partition.

ERROR MESSAGES

ERROR MESSAGE	MEANING
No MS-DOS partition. Use FDISK to correct (FORMAT)	An attempt has been made to format a hard disk that does not have an MS-DOS partition.
Non-DOS disk (device error)	The file allocation table contains invalid information. The disk must be reformatted.
No object modules specified (LINK)	You have not specified any object modules for the linker
No operating system on fixed disk (FDISK)	Self-explanatory.
No paper (device error)	The printer is either not switched on or is out of paper.
No path (MS-DOS)	You typed PATH to display your search path. There is no current command search path.
No room in directory for file (EDLIN)	The directory of the specified diskette is already full, or the specified disk drive or file name is illegal.
No space for a XXX cylinder partition at cylinder YYYY (FDISK)	There is not enough space on disk to accommodate a partition with the specified number of cylinders at the specified position.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Non-system disk or disk error. Replace disk and strike any key. (Startup)	Restart the system with another MS-DOS disk in drive A.
Not a block device (FORMAT)	You are trying to format a serial device. You cannot format serial devices.
Not enough memory (JOIN, SHARE) (Other External Commands)	There is not enough memory for MS-DOS to perform the command.
Not enough room for MS-DOS on this disk (SYS)	There is not enough room on the target disk for SYS to transfer the system files.
Not enough room to merge the entire file (EDLIN)	There was not enough room in memory to enable a Transfer command to merge the entire contents of a file.
Not found (EDLIN)	The search string was not found in the specified line range, or no further oc- currences were found after resuming.
Not ready (PRINT)	If this message occurs when PRINT at- tempts a disk access, PRINT will keep trying until the drive is ready. Any other error causes the current file to be cancelled. An error message would be output on your printer in such a case.

ERROR MESSAGE	MEANING
Not ready (device error)	The named device is not available for the read/ write operation required.
Out of environment space (MS-DOS)	There is not enough room in the program environment to accept more data.
Out of memory (MS-DOS and commands)	Self-explanatory.
Out of space on list file (LINK)	There is not enough space on disk to hold the list file.
Out of space on run file (LINK)	There is not enough space on disk to hold the run file.
Out of space on VM.TMP (LINK)	There is not enough space on disk to allow the VM.TMP file to be expanded.
Parameters not compatible (FORMAT, REPLACE)	You have specified switches which cannot be used together.
Parameters not compatible with fixed disk (FORMAT)	The FORMAT command was called with parameters which are only applicable to floppy diskettes.
Parameters not supported (FORMAT)	You have specified parameters that FORMAT does not support.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Parameters not supported by Drive (FORMAT)	Format displays this message when the device driver for this drive does not support Generic IOCTL function requests.
Path not found (PRINT)(MS-DOS)	You specified an invalid pathname.
Pathname too long (PRINT)(TREE)	The maximum pathname permitted is 63 characters.
PRINT queue is empty (PRINT)	There are no files waiting to be printed.
PRINT queue is full (PRINT)	You have tried to place more than ten files in the print queue.
Printer error (MODE)	<p>The printer mode has not been set because of one of the following conditions:</p> <ul style="list-style-type: none">• I/O error• Out of paper• Power off• Time out

ERROR MESSAGE	MEANING
<p>Probable non-DOS disk. Continue (Y/N)? (CHKDSK)</p>	<p>The disk you are using is not recognized by this version of MS-DOS. The disk either was created by another system with a format that is not supported on this version of MS-DOS or is not an MS-DOS disk. Do not continue processing if CHKDSK returned this message for a removable disk. If this message is returned for a hard disk, the information describing the characteristics of the disk to MS-DOS has been destroyed. In this case, you may continue CHKDSK processing.</p>
<p>Processing cannot continue (CHKDSK)(FDISK)</p>	<p>There is not enough memory in your machine to process CHKDSK for this disk. You must obtain more memory to run CHKDSK.</p>
<p>Program too big to fit in memory (MS-DOS)(LINK)</p>	<p>You must acquire more memory to run your application. It is possible that some programs you have run are still using some memory. Try to restart MS-DOS, and retype the command. If you still receive the error message and have used the <code>BUFFERS = parameter</code> directive in the systems CONFIG.SYS file reduce the number of buffers, re-boot the system and retry the command. However, if you still receive this message, you must acquire more memory.</p>

ERROR MESSAGES

ERROR MESSAGE	MEANING
Read fault (device error)	MS-DOS cannot read the requested data from the named device.
Read error in: filename (Commands)	The command could not read the entire file.
Requested drive is not available (FORMAT)	Self-explanatory.
Requested stack size exceeds 64K (LINK)	You have tried to specify a stack size greater than 64 Kbytes.
Rom BASIC not available Press reset to re-boot (BIOS)	<p>If this message occurred when you tried to boot from hard disk, the most likely cause is that there is no active partition on the hard disk. Bootstrap from floppy disk, run FDISK to make the bootable partition active.</p> <p>If this message occurred when you are running a program, that program may have been trying to access Rom BASIC or a spurious soft interrupt 18 hexadecimal occurred.</p>
Sector not found (device error)	The sector containing the data cannot be found, usually due to a defective area on the disk.

ERROR MESSAGE	MEANING
Sector size too large in file filename (Startup)	The device sector size defined in the device driver filename exceeds the system limit.
Seek (device error)	The disk drive cannot find the proper track on the disk.
Segment size exceeds 64K (LINK)	You have tried to combine identically named segments resulting in a segment requirement of more than the addressing limit of 64 Kbytes.
SHARE already installed (SHARE)	Share can only be installed once.
Sharing buffer exceeded (device error)	The memory space allocated by the command SHARE is insufficient. Rebootstrap your computer and call SHARE with a larger memory space parameter. If the message re-occurs, continue to increase the size of the memory space parameter.
Sharing Violation (device error)	A program tried to access a file, but another program is using that file. Press A to Abort, or wait a while and press R to Retry.
Source and target drives are the same (BACKUP) (RESTORE)	Specify a different source and target drive in your BACKUP or RESTORE command.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Source does not contain backup files (RESTORE)	The diskette in the drive was not created by BACKUP. The first file is not called BACKUP.@@@.
Source path required (REPLACE)	You did not specify a source path parameter.
Specified COMMAND search directory bad [access denied] (MS-DOS)	The SHELL directive in the CONFIG.SYS file is incorrect. The place that you have told MS-DOS to find COMMAND.COM does not exist, or COMMAND.COM is not in that place, or access has been denied to the directory containing COMMAND.COM.
Specified drive does not exist, or is nonremovable (DISKCOMP) (DISKCOPY)	If you referred to your hard disk drive as a parameter of these commands, you will get this error message. Otherwise you are referring to a non-existent drive.
Stack size exceeds 65535 bytes (LINK)	The specified stack size exceeds the system limit.
Symbol defined more than once (LINK)	Two or more modules have defined the same symbol name.
Symbol table capacity exceeded (LINK)	Very many, and/or very long names were entered exceeding the limit of approximately 25 Kbytes.

ERROR MESSAGE	MEANING
Syntax error (MS-DOS)	You have entered a command in the incorrect format.
System Failure (MODE)	Self explanatory.
SYS cannot install MS-DOS on this disk (SYS)	You cannot use SYS as the target disk either is not empty or was not formatted with the /S option.
Target cannot be used for backup (BACKUP)	BACKUP was called with /A parameter, but the target has not already been used as a backup disk.
Target is full (RESTORE)	RESTORE cannot continue for this reason.
Terminate batch job (Y/N)? (MS-DOS)	If you press CTRL BREAK or CTRL C while in batch mode, MS-DOS asks you whether or not you wish to end batch processing. Press Y to end processing. Press N to continue the batch job.
The last file was not restored (RESTORE)	If there is not enough room on the target disk to receive a particular backed up file, RESTORE will delete the partially backed up file. Delete unwanted files on the target disk, then enter RESTORE filename, where filename is the name of the file previously partially restored. Interruption to RESTORE can be continued in the same fashion.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Too many external symbols in one module (LINK)	There are too many external symbols in one module. The limit is 256 external symbols per module.
Too many groups (LINK)	Too many groups are defined. The limit is 10.
Too many libraries specified (LINK)	Too many libraries have been named. The limit is eight libraries.
Too many files open (MS-DOS)(EDLIN)	Try to solve this problem by increasing the FILES directive in CONFIG.SYS.
Too many overlays (LINK)	The system limit of the overlays has been exceeded
Too many public symbols (LINK)	There are too many public symbols in one module. The limit is 1024 public symbols.
Too many segments or classes (LINK)	You have too many segments or classes in your source files. The limit is 256 (segments and classes taken together).
Top level process aborted, cannot continue (COMMAND.COM) (Commands)	An unrecoverable Abortion occurred, when you chose ABORT in response to a disk error. The operating system has crashed. You must bootstrap MS-DOS from a new system disk.

ERROR MESSAGE	MEANING
Tree past this point not processed (CHKDSK)	Track 0 of the disk being checked is damaged. CHKDSK cannot continue.
Unable to copy keyboard routine (SELECT)	SELECT has a problem opening or reading the KEYBXX.COM file on your source disk. Run CHKDSK on the file to determine the cause of the problem
Unable to create directory (MKDIR)	<p>The specified directory cannot be created due to one of the following conditions:</p> <ul style="list-style-type: none"> • Directory already exists. • One of the directory path-names could not be found. • The root directory is full.
Unable to shift Screen (MODE)	Mode is unable to shift the test pattern on the screen any farther.
Unable to write BOOT (FORMAT)	The first track of the diskette, or of the hard disk partition, specified is bad. If it is a diskette problem, use another diskette. If it is hard disk problem, rerun FDISK specifying another starting track for the disk partition.
Unexpected DOS Error <i>nnn</i> (REPLACE)	An internal MS-DOS error has occurred. The number <i>nnn</i> is the internal MS-DOS error number. See the MS-DOS System Programmer Guide for a complete list of MS-DOS error numbers.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Unexpected end-of-file on library (LINK)	There is probably an error in the library file.
Unexpected end-of-file on VM.TMP (LINK)	The diskette containing VM.TMP is not present in the drive.
Unrecognized command in CONFIG.SYS	You have entered an invalid command in the configuration file.
Unrecoverable error in directory. Convert directory to file (Y/N)? (CHKDSK)	If you press Y in response to this prompt, CHKDSK will convert the bad directory into a file. You can then fix the directory yourself or delete it. If you press N , you may not be able to write to or read from the bad directory.
Unrecoverable file sharing error (SHARE)	Caused by a file-sharing conflict.
Unrecoverable read error on drive d: (DISKCOMP) (DISKCOPY)	Self-explanatory. The diskette has probably been damaged.
Unresolved externals: list (LINK)	The external symbols listed were not defined in the modules or library files. Do not attempt to run the file created by the linker.
VM.TMP is an illegal filename and has been ignored (LINK)	You cannot use VM.TMP as an object module.

ERROR MESSAGE	MEANING
Warning: directory full (RECOVER)	No more files can be recovered in the directory.
Warning-Diskette may be unusable (DISKCOPY)	After an unrecoverable read, write or verify error, the copy may be corrupted.
Warning: diskette is out of sequence Replace diskette or continue. Strike any key when ready. (RESTORE)	Self-explanatory.
Warning: The file above is marked read-only. Replace the file (Y/N)? (RESTORE)	When RESTORE /P is specified and the file encountered is read-only; answer Y if you want to replace the file or N if not. When you type ENTER , RESTORE will continue.
Warning: The file above was changed after it was backed up. Replace the file (Y/N)? (RESTORE)	When RESTORE /P is specified and the file on the target disk has a later time and date than the same named file on the source disk; answer Y if you want to replace the file on the target disk or N if not. When you type ENTER , RESTORE will continue.
Warning: no files were found to backup (BACKUP)	Check your backup file specifications for incorrect input.

ERROR MESSAGES

ERROR MESSAGE	MEANING
Warning: No files were found to restore (RESTORE)	Self-explanatory.
Warning: no stack segment (LINK)	None of the object modules specified contain a statement allocating stack space, but you entered the /STACK switch.
Warning: read error on EXE file (EXE2BIN)	The input file has not been read correctly.
Warning: Segment of absolute or unknown type (LINK)	There is a bad object module or an attempt has been made to link modules that the linker cannot handle; for example, an absolute object module.
Write error in TMP file (LINK)	No more disk space remains to expand the VM.TMP file.
Write error on RUN file (LINK)	There is not enough disk space for the Run file.
Write fault (device error)	MS-DOS cannot successfully write data from/to the named device.
Write protect (device error)	You have tried to write data to a disk that is write-protected.
Write protected (DISKCOPY)	You cannot use a write protected diskette for a copy.

3. GLOSSARY OF TERMS

E. GLOSSARY OF TERMS

ABOUT THIS APPENDIX

This appendix describes some technical terms, whose understanding is necessary for the user of MS-DOS.

CONTENTS

GLOSSARY OF TERMS

E-1

GLOSSARY OF TERMS

GLOSSARY OF TERMS

The following table defines the terminology in this manual.

TERM	MEANING
active partition	The partition on hard disk which contains the operating system files enabling the bootstrapping of the computer. This happens on system reset or when the computer is turned on.
ASCII	American Standard Code for Information Interchange. A 7 BIT code, which has been extended to an 8 BIT code (a BYTE) to represent graphic characters and international characters.
basic input output system (BIOS)	Part of the operating system which provides an interface with the machine hardware. Most of the BIOS is in Read Only Memory (ROM), the rest is loaded from the system disk.
binary digit (BIT)	In a binary numbering system, only two marks are used 0 and 1. Each of these marks is called a binary digit.
bootable file	A file of a specific format that the bootstrap loader can load into memory to initialize the system.
byte	Eight bits, which is normally a code for an ASCII character.
current directory	The directory in which you are working.

TERM	MEANING
cylinder	Hard disks usually consist of a number of platters. A cylinder refers to the same track on each surface of the platters which form a notional cylinder.
disk	A diskette or hard disk.
diskette	A single or double-sided 5 1/4 in. floppy disk.
drive specifier	A letter referring to the diskette drive or hard disk drive in question. For example it may be: A - first diskette drive. B - second diskette drive. C - hard disk drive.
editing function keys	The keys that invoke the intra-line commands.
external command	A command that is not loaded into memory at initialization. Such commands reside on disk from where they are loaded, executed and then purged from memory.
formatting	Disks must be formatted before they can be used with MS-DOS. Formatting places tracks, which are split into sectors, onto the surface(s) of a disk. The sectors are all the same length, typically 512 bytes. Also formatting places a boot record and an empty directory on the disk.

GLOSSARY OF TERMS

TERM	MEANING
hard disk	A sealed storage unit with non-removable surfaces. A hard disk can store much more information than a floppy disk, and the computer can retrieve information from it faster.
hardware reset	A system reinitialization caused by pressing the physical reset button. The subsequent initialization includes diagnostic tests and a reset of all system parameters. Any AUTOEXEC.BAT file or CONFIG.SYS file is executed.
inter-line commands	The EDLIN commands that operate on entire lines of text.
intra-line commands	The commands invoked by the special editing function keys that perform editing operations within a single line of text.
internal command	A command that is embedded in the COMMAND.COM file and resides in memory whenever MS-DOS is booted.
Kilobyte KB	$2 \text{ to the power } 10 = 1024 \text{ Bytes}$
Mega-byte MB	$2 \text{ to the power } 20 = 1\,048\,576 \text{ Bytes}$
nil parameter	A parameter to a command where the parameter in question is not specified in the command line. The parameter therefore assumes a default value.

TERM	MEANING
partition	A certain number of cylinders of a hard disk, which have been set aside for the use of a particular operating system. That operating system treats the partition like a complete, but smaller, hard disk. The maximum partition size allowed for MS-DOS is 32 MB. The number of cylinders this corresponds to, depends on how many bytes there are per cylinder.
pathname	A sequence of one or more directory names separated by backslashes, optionally beginning with a drive specifier and optionally terminating in a file name. It specifies a path through a directory structure to access a file or directory.
sectors	The track on a disk is divided into sectors. MS-DOS disks are soft sectored. The number of sectors per track is typically 8, 9 or 15.
source line	A line of text containing either the last command line entered or the current line in a file being edited. It can be retrieved in whole or part or modified using the special editing function keys.
system file	An MS-DOS file that is present on the MS-DOS system diskette that contains system software. There are three such files: Two hidden files and COMMAND.COM.

GLOSSARY OF TERMS

TERM	MEANING
system reset	A system reinitialization caused by pressing the CTRL , ALT and DEL keys simultaneously. Any AUTOEXEC.BAT file or CONFIG.SYS file is executed.
text file	An ASCII file whose records are separated by CR/LF.
tracks per inch (t.p.i.)	A disk track is the circular locus of the head as the disk rotates. The head can be moved to the other tracks; they are concentric circles. A double density diskette has 48 t.p.i. A quad density disk has 96 t.p.i.
virtual disk	An emulation of backing store in Random Access Memory (RAM). It is faster than disk backing store, but the information on virtual disk is lost when the computer is turned off.
volume label	A name that can be assigned to a disk by the FORMAT command. It will subsequently be displayed in a directory listing, or by the VOL command.
wild card character	A special symbol used to represent any single character (?), or any string of characters (*).
working session	The time between booting MS-DOS and the next boot of MS-DOS or switch-off.

F. THE VIRTUAL DISK SYSTEM

ABOUT THIS APPENDIX

This appendix describes how to install VDISK.SYS.

CONTENTS

INTRODUCTION F-1

**INSTALLING
VDISK.SYS F-1**

INTRODUCTION

A virtual disk is part of Random Access Memory which emulates a backing store Disk. The VDISK.SYS is a device driver, which when installed, enables a virtual disk drive with the next available drive letter. For example if you have a two physical drive machine, with two physical drives "A:" and "B:", when VDISK.SYS is installed, you will have an extra drive C:. The only difference between virtual disk and real disk is that when you turn your machine off, the information on virtual disk will be lost. So remember to COPY all files you want to keep, from virtual disk to a real disk, before you turn your machine off.

INSTALLING VDISK.SYS

VDISK.SYS is a file included on your MS-DOS Diskettes. To install the VDISK console drive the following command must be placed in the CONFIG.SYS file.

DEVICE = VDISK.SYS [*disk-size*][*sector-size*][*entries*] [/E[: *max*]]

Where

SYNTAX ELEMENT	MEANING
<i>disk-size</i>	A decimal value declaring the virtual disk size in Kilo Bytes. The value be from 1 through the maximum free Random Access Memory on your computer. The default is 64 Kilo bytes.
<i>sector-size</i>	A decimal value declaring the sector size in bytes. The value may be 128, 256 or 512. The default is 128 bytes.

SYNTAX ELEMENT	MEANING
<i>entries</i>	A decimal number declaring the maximum number of directory entries required for files. One entry is used for a volume label. Three entries are used for each sub-directory. The value may be from 2 through 512. The default is 64 entries.
<i>/E</i>	This switch specifies that the VDISK driver uses "extended memory". MS-DOS can only directly address 640 KB. Random Access Memory installed over 640 KB is "extended memory". This option is only available on Intel 80286 based Personal Computers, do not use this switch for other Personal Computers.
<i>:max</i>	This is a parameter of the <i>/E</i> switch, specifying the maximum number of sectors (of sector-size) to be transferred to/from extended memory. The possible values are a decimal number in the range 1 through 8. The default value is 8.

For example place the following assignment in CONFIG.SYS.

```
DEVICE = VDISK.SYS 128 512 32
```

Reboot your computer and a message similar to the following will be displayed:

```
VDISK Version 3.20 virtual disk D:  
Directory entries adjusted  
Buffer size: 128 KB  
Sector size: 512  
Directory entries: 32
```

Refer to Appendix C for more details of CONFIG.SYS.

THE VIRTUAL DISK SYSTEM

Remarks

The following situations prevent VDISK from being installed.

- less than 64 KB free memory.
- using the /E switch with insufficient extended memory.

In these cases the following message is output on the screen.

VDISK not installed insufficient memory

VDISK.SYS might adjust the parameters you specified in the following ways:

PARAMETER	ADJUSTMENT
<i>entries</i>	<p>Rounded up: 32 bytes (per file entry) multiplied by entries (rounded up) equals a multiple of sector-size. This is so as not to waste space available for directory entries.</p> <p>Rounded down: one sector at a time times sector-size divided by 32 bytes, until there are sufficient sectors to hold the File Allocation Table, the directory entries and at least two sectors for files. When the number of sectors for directory file entries has been rounded down to one, the above error message is issued. In this case redeclare the VDISK parameters in CONFIG.SYS and reboot the system.</p>
<i>disk-size</i>	<p>Rounded down: so as to leave 64 KB Random Access Memory free. If this is not possible the above error message is issued. In this case you need to install more Random Access Memory in order to use VDISK.</p>

Remarks

You can install more than one virtual disk by placing several **DEVICE = VDISK.SYS** commands in your CONFIG.SYS. Each virtual disk takes the next available drive letter. If the next available drive letter for the virtual disk is "F:", place the command:

LASTDRIVE = F

before your **DEVICE** command in CONFIG.SYS. Refer to Appendix C for more details.

Warning

Using VDISK in extended memory with networking installed, may lead to problems of lost interrupts and corruption of the virtual disk. If you have such problems, try setting the *max* parameter of the /E switch to 1. If this does not cure the problems, then you cannot place the declaration "DEVICE = VDISK.SYS /E" in the CONFIG.SYS, when you are going to load networking.

G. INSTALLING NEW BLOCK DEVICES

ABOUT THIS APPENDIX

This appendix describes how to install a device driver using DRIVER.SYS.

CONTENTS

INTRODUCTION G-1

INSTALLING DRIVER.SYS G-1

INSTALLING NEW BLOCK DEVICES

INTRODUCTION

The DRIVER.SYS is a device driver, which when installed either: enables a new block device to be installed; or enables existing block devices to be referred to as a new logical block device with different characteristics.

If you have installed an external floppy disk drive, you can create a block device driver for this drive using DRIVER.SYS; this will be referred to by the next available drive letter. For example on a twin floppy disk drive machine, when DRIVER.SYS is installed, the extra external drive will appear as drive C:.

The following is an example of the use of DRIVER.SYS for referring to an existing floppy disk drive by another drive letter. If you have a computer with a 1.2MB floppy disk drive and a hard disk. Before installing DRIVER.SYS for the 1.2MB drive, the drive letters A: and B: refer to this drive. After installing DRIVER.SYS for this drive, the drive letter "D:" can refer to the floppy disk drive specifically for handling 360KB diskettes. So to copy files from a 360KB diskette to a 1.2MB diskette, you could enter the command:

```
COPY D:*. * A:
```

MS-DOS will prompt you to enter the respective diskettes.

INSTALLING DRIVER.SYS

DRIVER.SYS is a file included on your MS-DOS Diskettes. To install this driver the following command must be placed in the CONFIG.SYS file.

```
DEVICE = DRIVER.SYS /D drive-no [/C] [/F: form-factor ]
```

```
[/H: heads ][/N] [/S: sectors ][/T: tracks ]
```

Where

SWITCH	PARAMETER	MEANING
/D	<i>drive-no</i>	Specifies the physical drive number between 0 and 255. Floppy disk drives start at 0, hard disks start at 128.
/C		Disk change support is required.
/F	<i>form-factor</i>	Specifies the media format and kind to be supported: 0 320/360 KB 1 1.2 MB 2 720 KB (default) 3 8 inch single density 4 8 inch double density 5 Hard Disk 6 Tape Drive 7 Other
/H	<i>heads</i>	Specifies the number of heads on the disk drive. Its value can range from 1 to 99. The default is 2 heads.
/N		Specifies a non-removable block device such as a hard disk.
/S	<i>sectors</i>	Specifies the number of sectors per track. Its value can range from 1 to 99. The default is nine sectors per track.
/T	<i>tracks</i>	Specifies the number of tracks per side. Its value can range from 1 through 999. The default is 80 tracks per side.

INSTALLING NEW BLOCK DEVICES

Note

DEVICE = DRIVER.SYS can be repeated on several lines of the CONFIG.SYS file with different parameters.

Examples

To add an external 3 1/2 inch drive to your computer, include the following line in your CONFIG.SYS:

```
DEVICE = DRIVER.SYS /D:2 /F:2
```

To refer to a 1.2MB drive as drive A: for 1.2MB diskettes and as drive D: (if D: is the next free drive letter) for 360KB diskettes, include the following command in your CONFIG.SYS.

```
DEVICE = DRIVER.SYS /D:0 /F:0 /H:2 /S:9 /T:40
```


INDEX

A

ANSI escape sequences, B-1
ANSI.SYS device
(CONFIG.SYS), B-1, C-5
archive bit, 5-10
ASCII tables, A-1
ASSIGN command, 5-8
ATTRIB command, 5-6, 5-10
AUTOEXEC.BAT, 1-2, 4-13

B

BACKUP command, 5-12
batch processing, 1-2, 4-10
batch processing commands,
4-10
BREAK command, 5-16
BREAK command
(CONFIG.SYS), C-1
BUFFERS command
(CONFIG.SYS), C-2

C

capacities
diskettes, 1-3
changing directory, 3-13
CHDIR (CD) command, 3-11,
3-13, 5-17
checking disks and files, 3-14
CHKDSK command, 3-14, 5-19
CLS command, 5-23
COMMAND command, 5-23
command formats, 4-1
commands
ASSIGN, 5-8
ATTRIB, 5-6, 5-10
BACKUP, 5-12
BREAK, 5-16
CHDIR (CD), 3-11, 3-13, 5-17
CHKDSK, 3-14, 5-19
CLS, 5-23
COMMAND, 5-23

COMP, 5-27
COPY, 3-4, 5-6, 5-30
CTTY, 5-36
DATE, 5-38
DEBUG, 9-1
DEL, 5-41
DIR, 3-8, 5-6, 5-43
DISKCOMP, 5-45
DISKCOPY, 5-48
ECHO, 5-51
EDIT (video file editor), 6-1
EDLIN (line editor), 7-2
ERASE, 5-52
EXE2BIN, 4-7, 5-52
EXIT, 5-55
external, 4-6, 4-8
FC, 5-56
FDISK, 5-66
FIND, 4-16, 5-66
FOR, 5-70
FORMAT, 5-72
GOTO, 5-76
GRAFTABL, 5-78
GRAPHICS, 5-79
GWBASIC, 5-82
HEXDUMP, 3-6, 5-83
IF, 5-85
internal, 4-6
JOIN, 5-87
LABEL, 5-92
LINK (the linker), 4-7, 8-3,
8-11, 8-13
MKDIR (MD), 3-13, 5-94
MODE, 5-95
MORE, 4-16, 4-17, 5-103
PATH, 3-10, 5-104
PAUSE, 4-10, 5-105
PRINT, 5-6, 5-107
PROMPT, 5-111
RECOVER, 5-114
REM, 4-10, 5-115
REN, 5-116
REPLACE, 5-6, 5-118
RESTORE, 5-122

- RMDIR (RD), 3-14, 5-126
- SELECT, 5-127
- SET, 5-129
- SHARE, 5-5, 5-131
- SHIFT, 5-133
- SORT, 4-16, 5-135
- SUBST, 5-139
- SYS, 5-143
- TIME, 5-144
- TREE, 5-146
- TYPE, 3-6, 5-147
- VER, 5-149
- VERIFY, 5-149
- VOL, 5-150
- XCOPY, 5-6, 5-151
- COMP command, 5-27
- compatibility
 - diskette, 1-4
 - drive, 1-5
- COMSPEC, 5-25
- CONFIG.SYS
 - definition, C-1
- CONFIG.SYS commands
 - BREAK, C-1
 - BUFFERS, C-2
 - COUNTRY, C-3
 - DEVICE, C-5
 - DRIVPARM, C-6
 - FCBS, C-8
 - FILES, C-9
 - LASTDRIVE, C-10
 - SHELL, C-11
 - STACKS, C-12
- CONFIG.SYS devices
 - ANSI.SYS, B-1, C-5
 - DRIVER.SYS, G-1, C-5
 - VDISK.SYS, F-1, C-5
- control keys, 2-1
- Control-break, 4-9
- Control-C, 4-9
- Control-Number Lock, 4-9
- Control-S, 4-9
- COPY command, 3-4, 5-6, 5-30

- COUNTRY command (CONFIG.SYS), C-3
- creating a new directory, 3-13
- CTTY command, 5-36
- cursor movement (ANSY.SYS), B-1

D

- DATE command, 5-38
- DEBUG command, 9-1
- debugging
 - commands, 9-4
 - definition of terms, E-1
- DEL command, 5-41
- DEVICE command (CONFIG.SYS), C-5
- device driver
 - definition, G-1
- device errors, D-1
- device names, 3-4
- DIR command, 3-8, 5-6, 5-43
- directories
 - changing, 3-13
 - creating, 3-13
 - definition, 3-6
 - displaying, 3-11
 - hierarchical, 3-6
 - removing, 3-14
 - root, 3-7
 - sub-directories, 3-6
- disk
 - fixed (hard), 1-3
 - hard (fixed), 1-3
 - virtual, 1-9, F-1
- disk operating system, 1-1
- DISKCOMP command, 5-45
- DISKCOMP command, 5-48
- diskette
 - capacities, 1-3
 - compatibility, 1-4
- diskettes, 1-3
 - handling, 1-5
 - labeling, 1-6

- purchasing, 1-6
 - write-protection, 1-6
- disks, 1-3
- displaying the current directory, 3-11
- drive

- compatibility, 1-5
 - identification, 1-3
- DRIVER.SYS device (CONFIG.SYS), C-5, G-1
- installation, G-1
- drives

- High-Capacity, 1-4
 - Normal-Capacity, 1-4
- DRIVPARM command (CONFIG.SYS), C-6

E

- ECHO command, 5-51
- EDIT command (video file editor), 6-1
- editing function keys, 2-4
- EDLIN (line editor) command, 7-2
- ENTER key, 2-1, 4-9
- Environment, 5-26
 - size, 5-26
- ERASE command, 5-52
- error level, 5-14, 5-55, 5-86 5-121
- error messages (MS-DOS), D-1
- examining files, 3-6
- EXE2BIN command, 4-7, 5-52
- exit code, 5-14, 5-55, 5-86 5-121
- EXIT command, 5-55
- external command syntax, 4-8
- external commands, 1-2, 3-10, 4-6

F

- FC command, 5-56

- FCBS command (CONFIG.SYS), C-8
- FDISK command, 5-66
- filenames
 - definition, 3-1

- files
 - examining, 3-6
 - naming, 3-1
 - protecting, 3-5
 - validity, 3-14

- FILES command (CONFIG.SYS), C-9
- FIND command, 4-16, 5-66
- fixed (hard) disk, 1-3
- font tables, A-1
- FOR command, 5-70
- FORMAT command, 5-72

G

- GOTO command, 5-76
- GRAFTABL command, 5-78
- GRAPHICS command, 5-79
- GWBasic command, 5-82

H

- hard (fixed) disk, 1-3
- HEXDUMP command, 3-6, 5-83
- hierarchical directories, 3-6

I

- IF command, 5-85
- input, 4-15
- input redirection, 4-16
- internal commands, 1-2, 4-6

J

- JOIN command, 5-87

K

Key reassignment
(ANSI.SYS), B-1
keyboard code tables, A-1

L

LABEL command, 5-92
LASTDRIVE command
(CONFIG.SYS), C-10
line editor, 7-1
 commands, 7-5
 keyboard functions, 7-36
LINK (the linker) command,
 4-7, 8-3, 8-11, 8-13
linker
 classes, 8-2
 groups, 8-2
 segments, 8-2
 VM. TMP, 8-1

M

MKDIR (MD) command, 3-13,
 5-94
MODE command, 5-95
MORE command, 4-16, 4-17
 5-103
MS-DOS commands, 4-8

N

national characters, A-1
networking, 5-5
notation conventions, 1-9

O

output, 4-15
output redirection, 4-15

P

parameters
 definition, 4-1
 nil, 4-5
PATH command, 3-10, 5-104
pathnames
 definition, 3-8
 maximum length, 3-10
 syntax, 3-9
PAUSE command, 4-10, 5-105
piping, 4-17
PRINT command, 5-6, 5-107
processor
 child, 5-26, 5-55
 parent, 5-26, 5-55
 secondary, 5-26, 5-55
programming
 FORTRAN, 1-1
 GW-BASIC, 1-1
 Macro Assembler, 1-1
 PASCAL, 1-1
PROMPT command, 5-111
protecting files, 3-5

R

read-only attribute, 5-10
real-time clock, 1-2
RECOVER command, 5-114
redirecting input, 4-16
redirecting output, 4-15
REM command, 4-10, 5-115
removing a directory, 3-14
REN command, 5-116
REPLACE command, 5-6,
 5-118
replaceable parameters, 4-14
 5-133
reserved device names, 3-4
RESTORE command, 5-122
RMDIR (RD) command, 3-14
 5-126
root directory, 3-7

S

- screen handling (ANSI.SYS),
B-1
- SELECT command, 5-127
- SET command, 5-129
- SHARE command, 5-5, 5-131
- SHELL command
(CONFIG.SYS), C-11
- SHIFT command, 5-133
- SORT command, 4-16, 5-135
- source drive, 4-9
- STACKS command
(CONFIG.SYS), C-12
- sub-directories, 3-6
- SUBST command, 5-139
- SYS command, 5-143

T

- target drive, 4-9
- TIME command, 5-144
- TREE command, 5-146
- TYPE command, 3-6, 5-147

V

- VDISK.SYS device
(CONFIG.SYS), F-1, C-5
- VER command, 5-149
- VERIFY command, 5-149
- video file editor, 6-1
- virtual disk, 1-9
 - definition, F-1
 - installation, F-1
- visual editor
 - commands, 6-21
 - keyboard functions, 6-4
 - searching, 6-19
- VOL command, 5-150

W

- Wild cards, 3-2

- the * wild card, 3-3
- the ? wild card, 3-2

X

- XCOPY command, 5-6, 5-151

NOTICE

Ing. C. Olivetti & C., S.p.A. reserves the right to make any changes in the product described in this manual at any time and without notice.

This manual is licensed to the Customer under the conditions contained in the User License enclosed with the Program to which the manual refers.

Code 4034460 Z (0)
Printed in Italy



olivetti